



Standard Product Interconnect

NOT TO BE DISCLOSED

Copy No. : 168 168
Personal Copy of: Mrs. Baeker

This specification has been developed jointly by Siemens and Philips and all rights are reserved by these two companies.

Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved even in the event of the grant of a patent or the registration of a utility model or design.



STANDARD PRODUCT INTERCONNECT

for

COMPATIBILITY OF DIGITAL IMAGING

NOT TO BE DISCLOSED

SPI Release 1

Date: 1987-12-01

Valid Documents List:

Doc.No.	Title	Version	Edition
1	Terminology	1	1
2	Concepts and Requirements	1	1
3	Application Services	1	1
4	Data Object Formats	1	1
5	Communication Interconnect	1	1
6	Off-Line Media and Data Formats	1	1
	Global Index	1	1

This specification has been developed jointly by Siemens and Philips and all rights are reserved by these two companies.

Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved even in the event of the grant of a patent or the registration of a utility model or design.



STANDARD PRODUCT INTERCONNECT

for

COMPATIBILITY OF DIGITAL IMAGING

NOT TO BE DISCLOSED

SPI Document 1

Terminology

Document Version No. 1 Date: 1987-12-01
Document Edition No. 1 Date: 1987-12-01

This document is part of the SPI Release defined in the current "SPI Valid Documents List".

The document consists of 16 pages excluding this cover page.

Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved even in the event of the grant of a patent or the registration of a utility model or design.

CONTENTS

1.0	TERMINOLOGY	2
1.1	GLOSSARY OF TERMS	2
1.2	RELATIONSHIP OF PACS ENTITIES	7
2.0	THE SPI DOCUMENT SET	8
2.1	SCOPE OF THE DOCUMENTS	8
2.2	DOCUMENT NUMBERING	8
2.2.1	Version	8
2.2.2	Edition	9
2.2.3	Release	9
3.0	SPI NOTATION CONVENTIONS	10
3.1	INTRODUCTION	10
3.2	SYNTAX NOTATION	10
3.3	OTHER CONSIDERATIONS	11
3.3.1	Conventions	11
3.3.2	Global Index and Document Indices	11
3.3	BASIC DATA ELEMENTS	12
	INDEX	15

CHAPTER 1

TERMINOLOGY

1.1 GLOSSARY OF TERMS

In the definitions below, a quoted term or phrase usually means the definition of this term or phrase (or parts of the phrase) occurs elsewhere in the glossary. Technical reference titles and phrases denoting a specially-introduced concept are also quoted.

ACR
American College of Radiology.

ACR-NEMA Standard
"Digital Imaging and Communications", ACR-NEMA Standards
Publication No. 300-1985.

Application Commands
All communication is defined in terms of Application Commands that are based on ACR-NEMA commands.

Communication Interconnect
The standard for the lower communication levels of SPI. In terms of the ISO/OSI model, Communication Interconnect encompasses roughly layers 1 through 4, viz. Physical, Data Link, Network and Transport layers.

Configuration Service (CFS)
Manages the information exchange between "IEs" about their capabilities.

Data Element
A subset of data, structured according to the ACR-NEMA keyword-length-value definition; uniquely identified by a "keyword" assignment. The same Data Element can be located in different data structures, since a "keyword" is part of each Data Element making it possible to identify and interpret each instantiation.

Data Object (DO)

A structured collection of "Data Elements" representing a particular entity. It has the following properties:

- It can be an SPI "Data Object", an ACR-NEMA "data set", or neither.
- Its length and type can always be found within it.
- Its type defines specific semantics and, possibly, coding.

All "data sets" (including Data Objects) have the same structure. The term "copy of a Data Object" refers to one instance of a Data Object at a particular location. All copies of a Data Object are identified with the same "UID".

Data Object Description (DESC)

A defined set of "Data Elements" that describes the elements expected to be queried for "Data Object" selection. The set of elements which constitute the DESC set is a function of Data Object type.

Data Set

"Data set" is used in accordance with its definition in the "ACR-NEMA Standard".

Directory Reference Table

A table on the "volume" that contains pointers to the "Primary" and "Secondary Key" directories on the "volume".

Export Service (EXS)

The capability of creating copies of "Data Objects" on removable media.

Group

As in ACR-NEMA, "A collection of 'Data Elements' which contain information of a similar nature."

High Level Functions (HLF)

Describe the standardized operations allowed in SPI.

Identifier, SPI

A string of printable ASCII characters used to name an SPI-defined item.

Identifier, Unique (UID)

An SPI "identifier" used to make a "Data Object" and all copies of the "Data Object" unique amidst other "Data Objects" within the SPI domain. All copies of a "Data Object" are identified with the same UID.

Image

A modality-specific and application-specific "pixel" data structure used by "Imaging Equipment".

Image Exchange Format

The logical data format used for "image" exchange and storage.

Image Management Service (IMS)

Keeps records of public (IMS-known) "Data Objects" for retrieval and query.

Image Series

A set of "images" acquired during an examination with the majority of settings of the equipment unchanged.

Imaging Equipment (IE)

A "PACS" entity that produces and handles "images" (see 1.2). In SPI, the terms "Imaging Equipment" and "node" are synonymous.

Index Area

That area on a "volume" where general information is stored for "volume" identification, description and directory references.

Key, Primary (PKEY)

A set of elements used for top-level access to information on a "volume". It is a defined subset of the Off-Line Media Object Identifier (OMO-ID) that is part of the "Mandatory Directory" on the "volume". It identifies one "Data Object" or a sequence of "Data Objects" using higher order identification parameters (e.g., patient-oriented parameters).

Key, Secondary (SKEY)

Any key that is used to make up an "Optional Directory" on a "volume".

Keyword

An "identifier" that provides for the unique identification of a "Data Element". It can be used to find properties or attributes of the data. The notion of keyword corresponds to a group- and element-number pair in the "ACR-NEMA Standard". SPI defines additional keywords.

Mandatory Directory (MDIR)

A directory structure of an off-line medium that must be written on each "volume". It describes the relation between "Data Object" identification and the location of the "Data Object" on the "volume".

Message

A structured package of information in a defined format allowing communication across an interface.

Modality

Refers to particular medical "Imaging Equipment" where an "image" is produced. For example: CT, NM, MR, DR, DS and US (see 1.2).

Modality-Specific Viewing Station

See "Viewing Station"; with the restriction that only "images" from a particular type of "modality" can be displayed (see 1.2).

Modality-Specific Workstation

See "Workstation"; provided with "modality-specific" functions and the ability to handle "images" from a particular type of "modality" only (see 1.2).

Multi-Modality Viewing Station

See "Viewing Station"; provided with the capability to display "images" produced by different types of "modalities" (see 1.2).

Multi-Modality Workstation

See "Workstation"; with the capability to display and process "images" produced by different types of "modalities" (see 1.2).

NEMA

National Electrical Manufacturers Association, Washington, DC.

Network Interface Equipment (NIE)

An active device that adapts the SPI protocols to network protocols.

Network Interface Unit (NIU)

Another term for "NIE".

Object Directory

A directory structure containing a list of "identifiers" and pointers to "Data Objects". The Object Directory is accessed via the "Primary Key".

Object Identification

An "identifier" that identifies one particular "Data Object" uniquely.

Optional Directory (ODIR)

An optional part of a directory structure describing relations between specific searching criteria and related "Data Objects".

PACS

Picture Archiving and Communication System.

PACS Console

A particular type of "Imaging Equipment" in a "PACS" environment (see 1.2).

Primary Key

See under "Key, Primary".

Private Service (PRS)

The capability of receiving/sending "private" "Data Objects".

Public Storage Service (PBS)

The capability of storing "public" "Data Objects".

Secondary Key

See under "Key, Secondary".

Service

A set of capabilities that are provided by an "IE".

SPI Domain

A set of interconnected "IEs" that conform to SPI.

SPI Identifier

See under "Identifier, SPI".

SPI-Keyword

See under "Keyword".

SPI Volume Label

See under "Volume Label, SPI".

Standard Product Interconnect (SPI)

A set of specifications to provide communication between "PACS" entities and to exchange data via removable media.

Unique Identifier

See under "Identifier, Unique".

Viewing Station

A console on which an "image" can be retrieved and displayed and for which a minimal set of "image" processing functions is possible.

Volume

An aggregate data area organization of a medium. "WORM" optical disks usually have one volume per side of the optical disk. Magnetic tapes and cartridges are referred to as one volume per reel. Removable magnetic disk packs usually have one volume per pack.

Volume Label, SPI

Contained in the "Index Area"; identifies a particular "volume" and contains "volume" parameters.

Workstation

An "image" display console on which "images" can be retrieved, displayed and processed.

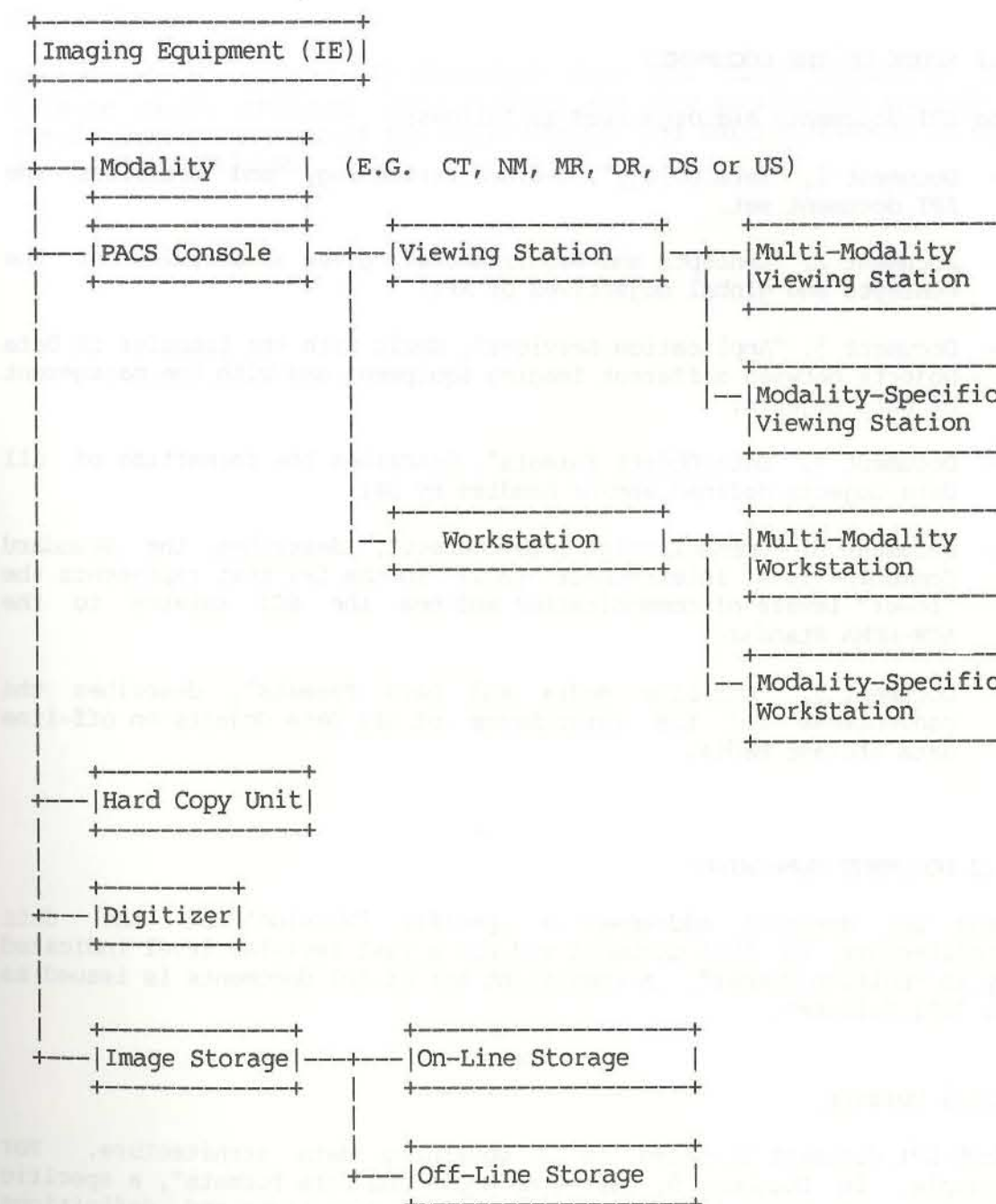
WORM

Write Once Read Many times: an abbreviation used in storage devices (e.g., for the non-erasable optical disk).

1.2 RELATIONSHIPS OF PACS ENTITIES

Figure 1.2-1, below, serves to relate the possible components and functional possibilities of Imaging Equipment.

Figure 1.2-1 Relationships of PACS Entities



CT = Computed Tomography, NM = Nuclear Medicine, MR = Magnetic Resonance, DR = Digital Radiography, DS = Digital Subtraction Angiography, US = Ultrasound

CHAPTER 2

THE SPI DOCUMENT SET

2.1 SCOPE OF THE DOCUMENTS

The SPI documents are organized as follows:

- Document 1, "Terminology", defines terminology and describes the SPI document set.
- Document 2, "Concepts and Requirements", gives an overview of the concepts and global objectives of SPI.
- Document 3, "Application Services", deals with the transfer of Data Objects between different Imaging Equipment and with the management of Data Objects.
- Document 4, "Data Object Formats", describes the formatting of all Data Objects defined and/or handled by SPI.
- Document 5, "Communication Interconnect", describes the Standard Communications Interconnect (SCI) in the SPI that represents the "lower" levels of communication and how the SCI relates to the ACR-NEMA Standard.
- Document 6, "Off-Line Media and Data Formats", describes the conventions for the interchange of SPI Data Objects on off-line data storage media.

2.2 DOCUMENT NUMBERING

Each SPI document addresses a specific "Version" of the data architecture in that document and has a text revision level indicated by an "Edition Number". A consistent set of SPI documents is issued as an "SPI Release".

2.2.1 Version

Each SPI document is based on a governing data architecture. For example, in Document 6, "Off-Line Media and Data Formats", a specific directory structure is defined. The set of data items and definitions in this document is defined to represent a Document 6 "Version Number".

2.2.2 Edition

A change to a document that does not affect the data architecture requires a change in its document revision level; a new "Edition Number" is assigned. Several editions of a document may cover the same "Version", but not the other way around. When the "Version Number" of a document changes, the "Edition Number" changes back to one.

2.2.3 Release

The sum of all relevant SPI documents form the "SPI Release". The Release needs changing only if there are very substantial changes in the documents. A list of the valid Version and Edition Numbers is kept for each SPI Release.

CHAPTER 3

SPI NOTATION CONVENTIONS

3.1 INTRODUCTION

This chapter provides precise definitions of SPI notation and data structures. As SPI is an extension of the ACR-NEMA Standard, the notations used by ACR-NEMA apply.

The basic number formats and character codes are defined in this chapter. Other basic element types are: character, string, array, keyword and derived elements like lengths, dates or names.

Other application-defined compounds of basic Data Elements, and structures composed of basic Data Elements that are not numbers, character strings or arrays, are considered to be non-basic or derived Data Elements.

SPI documents shall introduce and define specific data structures using the syntax and basic Data Elements defined below.

3.2 SYNTAX NOTATION

The following notation shall be used for syntax definitions:

- Terminal symbols in this syntax definition are written in single quotes: e.g., '0'.
- Non-terminal symbols are written between the bracket characters '<' and '>': e.g., <bit>.
- The metasymbol '::=' means: "is defined as".
- The '|' character separates alternatives.
- Optional parts are enclosed by the bracket characters '[' and ']'. Parts of a syntax definition may be grouped by means of the parenthesis characters '(' and ')'. A repetition factor shall be specified between the parenthesis characters '(' and ')', and shall directly follow the symbol it multiplies.

There are three forms of repetition:

1. (n) where 'n' is a numeric value means exactly 'n': e.g., <bit>(8).
2. (n) where 'n' is a letter means exactly 'n', 'n' being specified elsewhere: e.g., <bit>(n).
3. (m...n) means at least 'm', and at most 'n', where m < n, and 'm' and 'n' are as in case 1 or 2 above: e.g., <bit>(m...n).

- Spaces or new lines are ignored. A definition statement is terminated with the character ';'.

An example of a definition is given in Figure 3.2-1 together with two derivative interpretations. The primary definition means "element xxx is defined as the concatenation of aaa, optionally bbb and ccc, two times ddd with eee, and 1 to n times fff". The derivative interpretations show two ways of choosing options and repetition counts.

Figure 3.2-1 Sample Definitions

A primary definition:

```
<xxx> ::= <aaa> [<bbb> <ccc>] (<ddd> <eee>)(2) <fff>(1...n);
```

Two interpretations of <xxx>:

```
<aaa><ddd><eee><ddd><eee><fff><fff>
```

```
<aaa><bbb><ccc><ddd><eee><ddd><eee><fff>
```

3.3 OTHER CONSIDERATIONS

3.3.1 Conventions

Double quotes ("...") are normally used in the SPI documents; single quotes ('...') are used for special purposes in syntax definitions and inside text that is already quoted.

SPI ignores upper/lower case differences in ASCII strings used in command and Data Element text. The convention in other text is to capitalize commands and specific terminology for emphasis. Terms are not capitalized in general use. Thus, for example, "Data Object" and "Data Element" are capitalized, while "object" and "element" are not. A special case, "data set", remains uncapitalized (except in tables and special lists), in conformance with ACR-NEMA usage.

Dates that are not part of SPI-defined syntax are written in ISO-format, e.g., 24 September 1987 becomes 1987-09-24.

3.3.2 Global Index and Document Indices

The SPI Global Index combines the indices of the other documents. In all indices, locations of definition and significant discussion are marked with an asterisk appended to the page number or page number range. In the Global Index, the SPI document number is indicated for the first entry in the page list from each document.

Certain index entries are classified by adding a brief reference inside parentheses after the entry proper: "(DE)" designates Data Element; "(HLF)" designates High Level Function; "(ACR)" designates an ACR-NEMA term.

3.4 BASIC DATA ELEMENTS

In the text below, and in SPI Documents in general, the convention is adopted that a hexadecimal number is designated by appending the letter 'H' to the hexadecimal digit string. Numbers that are not designated this way are decimal unless otherwise indicated.

Data Element labels in the definitions below are not "case sensitive". Thus, "<date>" or "<Date>" or "<DATE>" (or other case transformations of the letters in the word "date") have the same meaning when used in other definitions.

<bit> ::= '0' | '1' ;

<byte> ::= <bit>(8) ;

<ascii-char> ::= <byte> ;

The byte contains a 7-bit ASCII character code occupying the low order part of the byte.

<alpha-char> ::= <ascii-char> ;

Permitted characters: 'A'...'Z' and 'a'...'z'.

The terminology "'A'...'Z'" indicates a sequence of ASCII characters between the first and last character, inclusive. This sequence is in the ASCII collating order. This notation is used elsewhere in the following text.

<decimal-char> ::= <ascii-char> ;

Permitted characters: '0'...'9'.

<sign-char> ::= <ascii-char> ;

Permitted characters: '+' and '-'.

<name-sep-char> ::= <ascii-char> ;

To be applied in person names.

Permitted characters:

Character	Value
dot (',')	"2EH"
dash ('-')	"2DH"
comma (',')	"2CH"
space (' ')	"20H"
apostrophe ('')	"27H"

<print-char> ::= <ascii-char> ;

Codes shall be for printable characters (ASCII "20H"..."7EH"). The space character ' ' ("20H") is printable; carriage return and line feed characters are not.

<integer-X> ::= <bit>(X) ;

Two's complement binary coded whole signed or unsigned number. The 'X' in the name equals the number of bits. For example, an 11 bit number can be described as <integer-11>. A variable length number will be: <integer-n> ::= <bit>(1...n); In this case, 'n' symbolizes the variable number that cannot be smaller than 2 for a signed number, and 1 for an unsigned number.

<A-int> ::= [<sign-char>] <decimal-char>(1...n) ;

Signed or unsigned whole number of variable length ASCII. When the sign character is missing, the value shall be interpreted as a positive number. The shortest possible string is a single decimal character.

<A-real> ::= <ascii-char>(n) ;

Floating point number, according to ANSI FORTRAN Standard; e.g., "+999.999E+99".

<A-number> ::= <A-int> | <A-real> ;

ASCII number that can be either an integer or a real number.

<Keyword> ::= <Group-nr> <Element-nr> ;

<Group-nr> ::= <integer-16>;

Always >= 0. Value shall conform to ACR-NEMA Standard.

<Element-nr> ::= <integer-16>;

Always >= 0. Value shall conform to ACR-NEMA Standard.

<Date> ::= <decimal-char>(4) <ascii-char '.'> <decimal-char>(2) <ascii-char '.'> <decimal-char>(2) ;

<Time> ::= <decimal-char>(2) <ascii-char ':'> <decimal-char>(2) <ascii-char ':'> <decimal-char>(2) <ascii-char '.'> <decimal-char>(0...n) ;

Format: hh:mm:ss.ss. (hours:minutes:seconds)
Seconds ("ss.ss") are significant to two decimal places.

<Name> ::= <alpha-char> (<alpha-char> | <name-sep-char>) (0...n) ;

Person name. Can be used for complete names and for family or first names separately. Full names shall precede initials and titles. There is no distinction between upper and lower case. <Name> is variable length.

<Ident> ::= <print-char>(n) ;

General identifier string. May contain any printable character.

<Alt> ::= <print-char>(n) ;

Format for a discrete type variable for which each value represents one of the possible alternatives. The values are coded as strings of printable characters.

<Enum-16> ::= <integer-16> ;

A 16 bit discrete variable containing a binary coded enumerated value. The values are in the range: "0000H"... "FFFFH".

<opt-info> ::= <opt-length> <opt-element-nr> <opt-element-data> ;

<opt-length> ::= <integer-16> ;

Number of bytes of the optional element (i.e., the number of bytes in the string composed of the optional element number together with the optional element data).

<opt-element-nr> ::= <integer-16> ;

Optional element number, always > 100H.

<opt-element-data> ::= <print-char>(n) ;

I N D E X

- A -

ACR 2*
ACR-NEMA Standard 2*
<A-int> 13*
<alpha-char> 12*
<Alt> 14*
<A-number> 13*
Application Command 2*
<A-real> 13*
<ascii-char> 12*

- B -

<bit> 12*
<byte> 12*

- C -

CFS = Configuration Service
Communication Interconnect 2*
Configuration Service 2*

- D -

Data Element 2*
Data Object 3*
Data Object Description 3*
<Date> 13*
<decimal-char> 12*
DESC = Data Object Description
Directory Reference Table 3*
DO = Data Object

- E -

Edition 9*
<Element-nr> 13*
<Enum-16> 14*
Export Service 3
EXS = Export Service

- F -

floating point number 13*

- G -

global index 11
Group 3*
<Group-nr> 13*

- H -

hard copy unit 7
hexadecimal number convention 12*
High Level Function 3*
HLF = High Level Function

- I -

<Ident> 14*
Identifier, SPI 3*
Identifier, Unique 3*
IE = Imaging Equipment
image 3*
exchange format 3*
series 4*
storage 7
Image Management Service 4*
Imaging Equipment 4*, 7
IMS = Image Management Service
Index Area 4*
<integer-X> 13*

- K -

Key, Primary 4*, 5
Key, Secondary 4*, 6
<Keyword> 13*
keyword 4*

- M -

Mandatory Directory 4*
MDIR = Mandatory Directory
Structure
message 4*
modality 4*, 7
Modality-Specific:
Viewing Station 4*, 7
Workstation 5*, 7
Multi-Modality:
Viewing Station 5*, 7
Workstation 5*, 7

- N -

<Name> 13*
<name-sep-char> 12*
NEMA 5*
Network Interface Equipment 5*
Network Interface Unit 5*
NIE = Network Interface Equipment
NIU = Network Interface Unit

- O -

Object Directory 5*
Object Identification 5*
ODIR = Optional Directory
Structure
off-line storage 7
on-line storage 7
<opt-element-data> 14*
<opt-element-nr> 14*
<opt-info> 14*
Optional Directory 5*
<opt-length> 14*

- P -

PACS 5*
PACS console 5*, 7
PBS = Public Storage Service
PKEY = Key, Primary
<print-char> 12*
Private Service 5*
PRS = Private Service
Public Storage Service 5*

- S -

Service 6*
<sign-char> 12*
SKEY = Key, Secondary
SPI:
= Standard Product Interconnect
document conventions 8*
document numbering 8*
document set 8*
domain 6*
Release 9*
terminology 2
Version 8*
Volume Label 6*
Standard Product Interconnect 6*
syntax 10*

- T -

terminal symbols 10*
<Time> 13*

- U -

UID = Identifier, Unique
Unique Identifier 3, 6

- V -

viewing station 6*, 7
volume 6*
Volume Label, SPI 6*

- W -

workstation 6*, 7
WORM = Write Once Read Many
Write Once Read Many 6*



STANDARD PRODUCT INTERCONNECT

for

COMPATIBILITY OF DIGITAL IMAGING

NOT TO BE DISCLOSED

SFI Document 2

Concepts and Requirements

Document Version No. 1 Date: 1987-12-01
Document Edition No. 1 Date: 1987-12-01

This document is part of the SFI Release defined in the current "SFI Valid Documents List".

The document consists of 23 pages excluding this cover page.

Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved even in the event of the grant of a patent or the registration of a utility model or design.

CONTENTS

1.0	INTRODUCTION	2
1.1	OBJECTIVES	2
1.2	BASIC SPI AND DATA OBJECT REQUIREMENTS	2
1.3	SCOPE OF THIS DOCUMENT	4
1.4	APPLICABLE DOCUMENTS	4
1.5	OTHER CONSIDERATIONS	4
2.0	SCOPE OF COMPATIBILITY	5
2.1	SPI GOALS	5
2.2	SPI ENVIRONMENT	7
2.3	IMPLEMENTATION OF SPI FUNCTIONALITY	9
3.0	CONCEPTUAL MODEL	11
3.1	OVERVIEW OF THE SPI	11
3.2	SPI STRUCTURE	12
4.0	REQUIREMENTS	15
4.1	APPLICATION SERVICES	15
4.1.1	Image Management Service (IMS)	15
4.1.2	Public Storage Service (PBS)	16
4.1.3	Private Service (PRS)	16
4.1.4	Export Service (EXS)	16
4.1.5	Configuration Service (CFS)	16
4.2	DATA OBJECTS	16
4.2.1	Classification of Data Objects	16
4.2.2	Location of Data Objects	16
4.2.3	Identification of Data Objects	16
4.2.4	State of a Data Object	17
4.2.5	Data Object Type	17
4.2.6	Protection	17
4.2.7	Commands	17
4.3	TRANSMISSION PROTOCOL	18
4.4	PHYSICAL LINK AND ELECTRICAL CHARACTERISTICS	18
4.5	OFF-LINE MEDIA FORMATS	18
5.0	DATA INTEGRITY	19
6.0	SERVICEABILITY	20
7.0	VERIFICATION	21
7.1	SCOPE OF VERIFICATION	21
7.2	TEST PROCEDURE	21
7.3	TEST CONFIGURATION	21
7.4	PROGRAM OF TESTS	22
	INDEX	23

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVES

The Standard Product Interconnect (SPI) is specified to standardize transfer of Data Objects between different Imaging Equipment (IE). The SPI is intended to support a Picture Archiving and Communication System (PACS) that contains various types of IEs and possibly NIEs. The goal of SPI is to obtain compatibility and the capability for different IEs, possibly of different manufacturers, to exchange medical images and associated information.

A fundamental premise in SPI is that each modality is allowed to retain as much autonomy as possible, consistent with overall system integrity requirements. Thus, the response of an IE to queries and/or requests to send Data Objects depends on its policies as well as its capabilities.

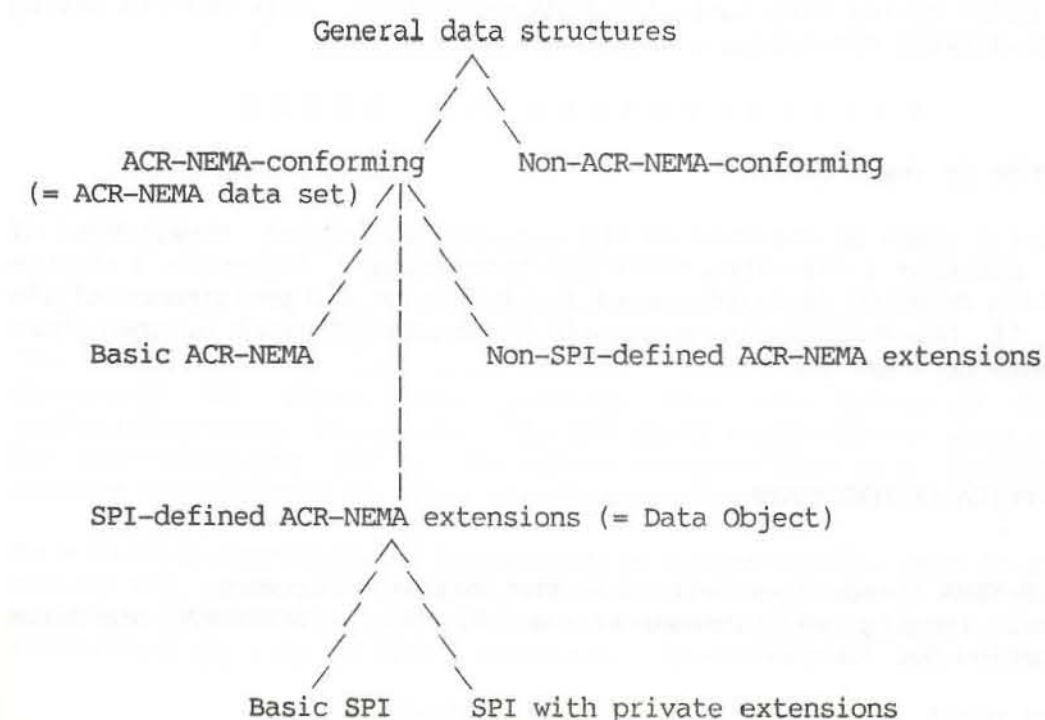
The definition of the SPI includes the basic methods and mechanisms used for the transfer of Data Objects, i.e., the Data Object requirements, the Data Object structure definition that provides the basis for all communication transactions, the transport protocol and services, the physical link, and the data formats of storage media.

1.2 BASIC SPI AND DATA OBJECT REQUIREMENTS

A Data Object is a structured collection of Data Elements that represents a particular entity. It is more fully described in 3.1.

Data structures used by IEs can be classified as shown in Figure 1.2-1 (page 3).

Figure 1.2-1 Data Structure Classification



The SPI standard has been defined insofar as possible as a valid extension of the ACR-NEMA Standard. The following requirements guided its formation:

- No SPI definition shall contradict the ACR-NEMA standard.
- The ACR-NEMA Minimum Requirements shall be met.
- SPI data formats shall satisfy ACR-NEMA-defined rules for data formats.
- Non-ACR-NEMA-defined data formats added by the SPI shall adhere to ACR-NEMA-defined rules for data formats.
- SPI shall allow other physical communication than ACR-NEMA.
- No ACR-NEMA-conforming Data Object shall be refused SPI services because of the presence of ACR-NEMA optional elements.
- SPI services may require that Data Objects have Data Elements that ACR-NEMA allows but does not require. Such required elements may be different for different services.
- No SPI implementation may refuse services because of the presence of SPI-optional Data Elements.

The SPI is intended to support not only SPI-conforming PACS nodes, but also nodes having only basic ACR-NEMA capability. Data Objects having non-SPI-defined extensions are also to be supported.

1.3 SCOPE OF THE DOCUMENTS

Document 2 gives an overview of the concepts and global objectives of SPI. Document 1 describes basic SPI terminology. Documents 3 through 6 contain detailed descriptions of the functions and performance of the SPI. If Document 2 disagrees with Documents 3 through 6, then those documents have precedence.

1.4 APPLICABLE DOCUMENTS

The ACR-NEMA Standard is defined in the following document:
"Digital Imaging and Communications Standard", ACR-NEMA Standards Publication No. 300-1985.

The OSI Model is defined in the following document:
"Information Processing Systems - Open Systems Interconnection - Basic Reference Model", International Organization for Standardization, ISO 7498, 1983.

1.5 OTHER CONSIDERATIONS

Double quotes ("...") are primarily used in the SPI documents; single quotes ('...') are used for special purposes in syntax definitions and inside text that is already quoted.

SPI ignores upper/lower case differences in ASCII strings used in command and Data Element text. The convention in other text is to capitalize commands and specific terminology for emphasis. Terms are not capitalized in general use. Thus, for example, "Data Object" and "Data Element" are capitalized, while "object" and "element" are not. A special case, "data set", remains uncapitalized (except in tables and special lists), in conformance with ACR-NEMA usage.

Dates that are not part of SPI-defined syntax are written in ISO-format, e.g., 24 September 1987 becomes 1987-09-24.

CHAPTER 2

SCOPE OF COMPATIBILITY

2.1 SPI GOALS

The general objectives for the use of the SPI are listed in Figure 2.1-1 (page 6). Specifically, inclusion of the SPI shall not change the local autonomy of interconnected IEs. IEs are defined and developed as stand-alone products that are optimized for the application which they serve. The SPI shall enable these products to be interconnected and to be integrated into high level systems with minimum interference to their stand-alone characteristics.

In a Picture Archiving and Communication System (PACS), each IE must be customized, e.g., for identification and user names. The SPI must permit a PACS to grow with minimum redesign to higher performance and functionality, e.g., higher transmission speed, and new protocols.

Figure 2.1-1 System Goals Supported by the SPI

<u>Intended for</u>	<u>Not intended for</u>
<u>System Scope</u>	
<ul style="list-style-type: none"> . Demand and prescheduled service. . Based on message transfer. . Interfacing computer-based products. . Communication of data between image generation, image storage, image presentation and image processing equipment. . Each node as a stand-alone system. . Low implementation effort to include SPI in products. . Optionally: compressed images. 	<ul style="list-style-type: none"> . Continuous-use, dedicated channels. . Wide-area or global communication. . Local data traffic internal to products. . Multidrop.
<u>Standardization Structure</u>	
<ul style="list-style-type: none"> . Independence between IEs and network. . Standard Data Object structure. . Standard protocols. . Open data format and length. . Use of computer industry standardization. . Accommodating ACR-NEMA data formats 	<ul style="list-style-type: none"> . Definition of the human-user functions in a Picture Archiving and Communication System (PACS).
<u>Flexibility/Versatility</u>	
<ul style="list-style-type: none"> . Wide range of IEs and networks. . Independence of application functions. 	<ul style="list-style-type: none"> . Image traffic only. . Particular network type. . Particular family of networks. . Fixed data format.
<u>Capacity</u>	
<ul style="list-style-type: none"> . No restriction relative to ACR-NEMA. 	
<u>Priority</u>	
<ul style="list-style-type: none"> . Message processing priority not defined by SPI. 	
<u>Reliability</u>	
<ul style="list-style-type: none"> . Error detection on SPI data link layer and retransmission of faulty frames. 	
<u>Security</u>	
<ul style="list-style-type: none"> . Dedicated application functions. 	<ul style="list-style-type: none"> . Protection against malicious users.

2.2 SPI ENVIRONMENT

The SPI supports point-to-point connections between Imaging Equipment (IE). Where the distance between IEs exceeds the limited ACR-NEMA physical link capability, various techniques, e.g., fiber optic cable, may be used to extend the range.

In an application involving several IEs, the communication could be through a network. Different types of IEs would be interfaced to the network using Network Interface Equipment (NIE). The communications environment of a network is beyond the scope of the definition of the SPI; see Figure 2.2-1 (page 8).

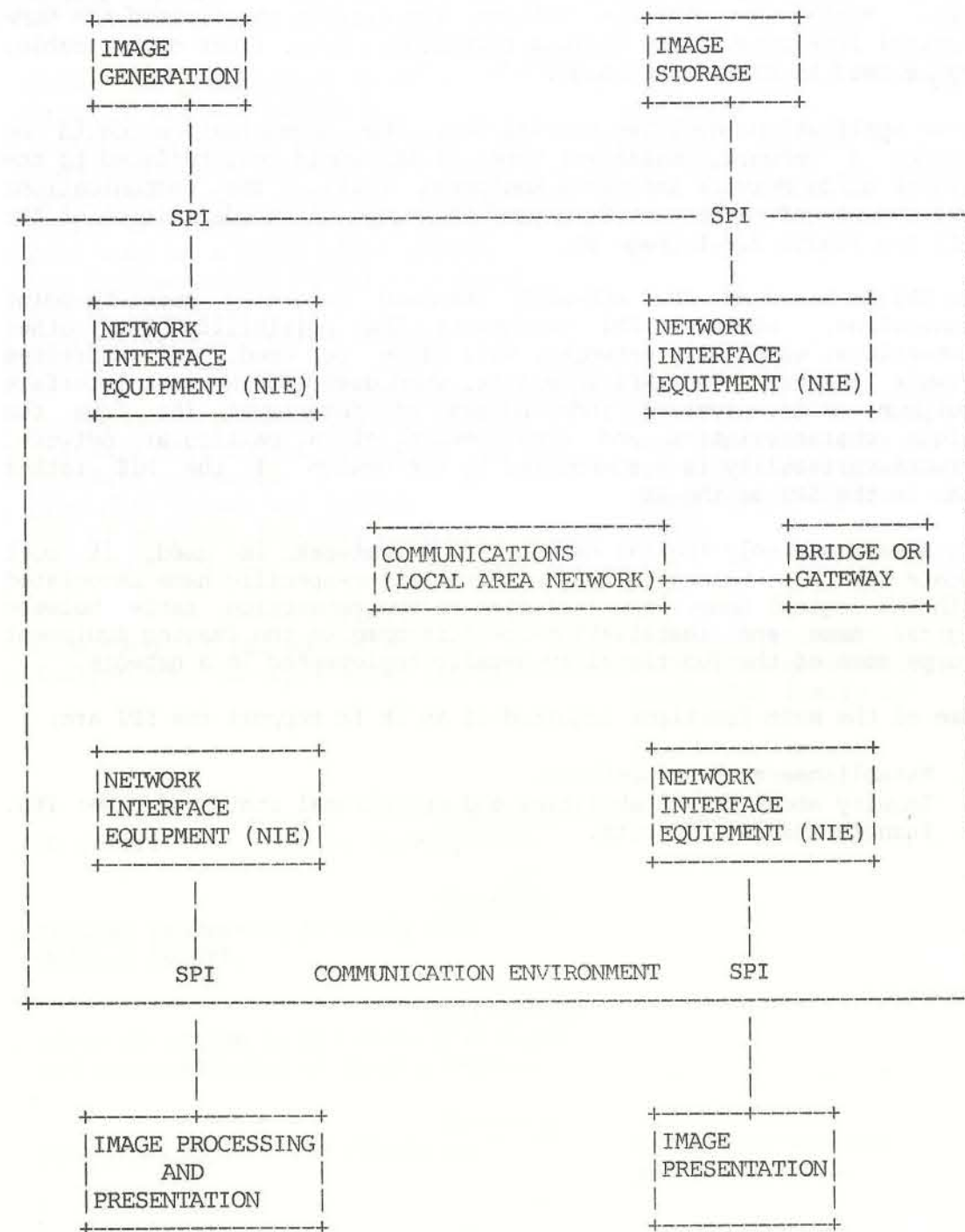
The SPI is based on the ACR-NEMA Standard governing point-to-point connections. However, SPI recognizes the possibility that other connections, especially networks, will often be used, and addresses network issues. The SPI interface, when used with Network Interface Equipment (NIE), provides independence of individual IEs from the unique characteristics and requirements of a particular network. Network variability is accommodated by the design of the NIE rather than in the SPI or the IE.

IEs always use only logical names. If a network is used, it must locate the IE belonging to the installation-specific name associated with the logical name. The inclusion of the conversion table between logical name and installation-specific name in the Imaging Equipment usurps some of the functionality usually implemented in a network.

Some of the main functions required of an IE to support the SPI are:

- Establishment of connections.
- Inquiry about the capabilities and operational status of other IEs.
- Transfer of Data Objects.

Figure 2.2-1 SPI Environment



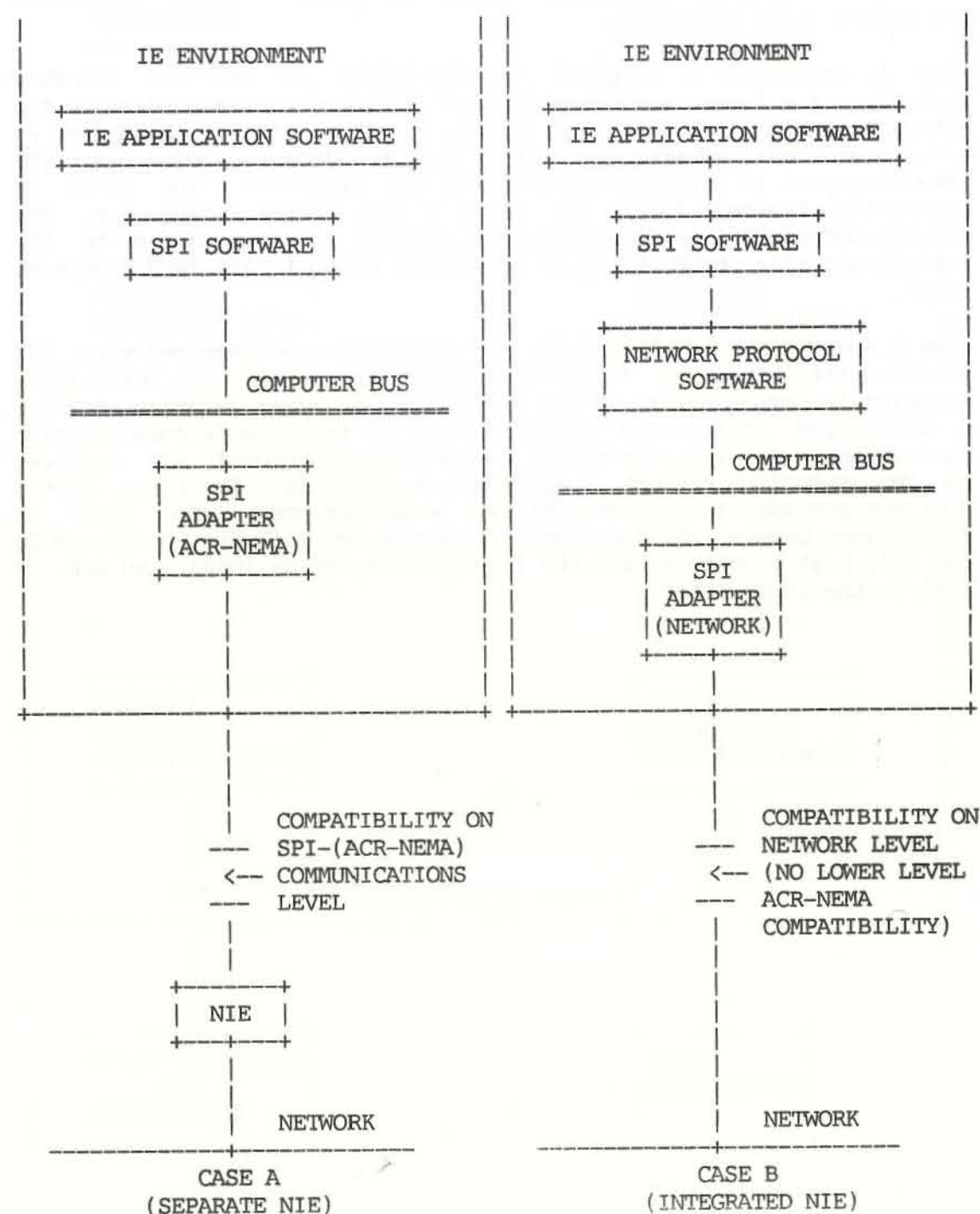
2.3 IMPLEMENTATION OF SPI FUNCTIONALITY

To match the requirements of the standard, IEs shall have the hardware and software for the SPI interface fully integrated into the equipment, or they shall employ a separate adapter to perform these functions. See Figure 2.3-1 (page 10).

Case A indicates a complete implementation of the SPI Standard including the lower-level communication protocols corresponding to the ACR-NEMA standard. One part of the SPI functionality is implemented on a separate SPI adapter connected to the IE's computer bus. The remaining part is performed by the IE's SPI software. The level of separation between these two parts is left to the implementor. The network-independent SPI communication protocol is transformed to the network-specific protocol by an additional Network Interface Equipment (NIE).

Case B assumes a network-specific SPI implementation combined with NIE functionality that is fully integrated into the IE. The integration achieves IE compatibility on the network level. This solution requires a formalized choice and specification of the network communication environment. The lower-level SPI communication protocols are replaced by the specific network communication protocols. Therefore, Case B does not provide full compatibility with ACR-NEMA. The level of separation between the functionality of the SPI software, performed by the IE's host processor, and the functionality of an integrated NIE is left to the implementor.

Figure 2.3-1 Implementation of SPI Functionality



CHAPTER 3

CONCEPTUAL MODEL

3.1 OVERVIEW OF THE SPI

Four key elements are basic to an understanding of SPI structure: message structure, Application Commands, transmission protocol and the physical link connections between IEs and/or NIEs.

MESSAGE STRUCTURE

The "SPI message" is the basic vehicle for communicating information from one IE to another. Each message consists of a command (Group 0, and optionally, Group 1) and an ACR-NEMA data set (which may be null). The data set is a coherent structure that can be used to encapsulate any type of information ranging from text to structures like patient information and digital images. The message that contains the data set contains information to allow accurate delivery, interpretation, and response. All data sets (including Data Objects) have the same structure. While the data set allows open definition of its use for particular applications, the SPI also defines Data Object standards for images and other types of data. The SPI standard for Data Object structure and formats is defined in Document 4; it is based on the ACR-NEMA definition for data set structure and formats (see 1.4).

APPLICATION COMMANDS

For an IE to use the SPI protocols and to be able to communicate its wishes and intentions with other peer IEs and applications, the SPI specifies a family of standard High Level Functions implemented by Application Commands that are intended to cover application requirements. The Application Commands and High Level Functions are described in Document 3.

TRANSMISSION PROTOCOL

The SPI provides multiple logical end-to-end connections between communicating application processes via virtual channels. The transmission protocol guarantees accurate message delivery without duplications. Multiple Data Objects can be sent in sequence over a connection once it has been established. Expedited delivery of brief control-type messages can be achieved ahead of long, image-containing messages already under way by establishing new simultaneous connections. The specific mechanisms of communication are transparent to the application process. Details of this communication function are described in Document 5.

PHYSICAL LINK

A standard physical link and its associated control can couple an IE to its NIE to provide independence from the particular type of local area network employed in a given installation. For communication between only two IEs over a limited distance, this link can operate without the NIE and local area network. The SPI standards for this link are defined in Document 5.

3.2 SPI STRUCTURE

The functions of the SPI are layered in conformance with the OSI Model and the ACR-NEMA Standard (see 1.4). This enhances portability between various applications and permits independent development and optimization of individual layers without affecting other layers.

Figure 3.2-1 (page 13) shows the correspondence between the OSI Model and the implementation of the SPI. A typical communication from IE A to IE B is initiated by IE A, is translated into standard message formats in the protocols of the Application and Presentation layers, and is passed down to the Transport layer. IE A establishes a connection with its counterpart Transport layer at destination IE B and breaks the message into smaller packets for communication across the connection. The lowest layers in IE A manage the transmission of these packets across the SPI link to the NIE that, in turn, manages the access and transmission across the local area network and its links to its counterpart NIE at the destination. There, the peer layers perform the reassembly of messages and deliver the data to the destination application process in the particular form it can understand. The long arrow of stars snaking around the lower part of Figure 3.2-1 (page 13) indicates the functional flow of information through the model.

A more detailed definition of the functions in the respective layers is given in Figure 3.2-2 (page 14). This figure is an overview of the SPI and specifies the key parameters of each layer and their inter-relationships.

Figure 3.2-1 SPI Conformance to the OSI Model

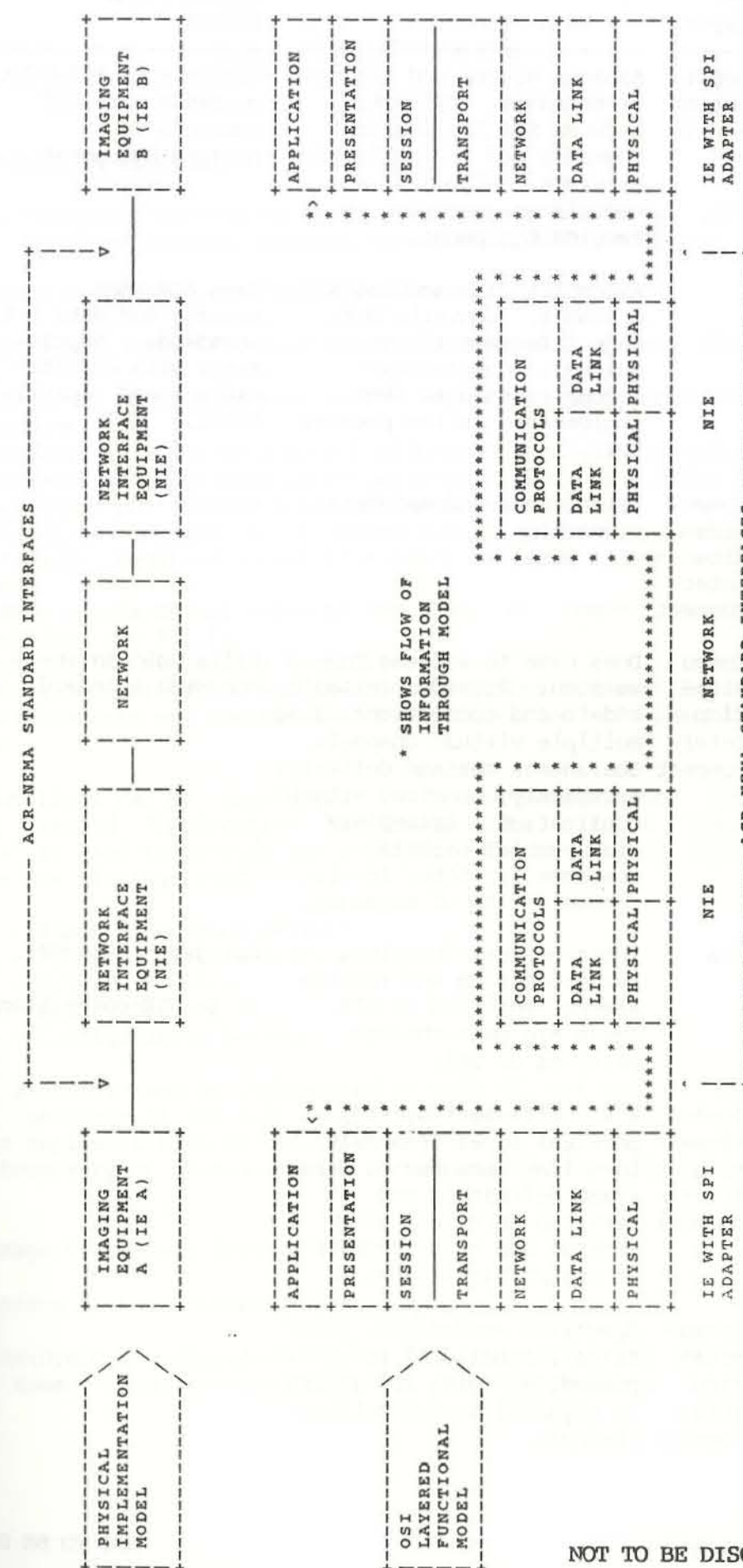


Figure 3.2-2 Overview of OSI Layers

OSI layer	SPI layer	Major Functions	Approach and Issues	SPI Doc No
Application	Application Services	Manages access and transfer of messages. Converts between SPI Application Commands and IE Application Services. Allocates and negotiates resources in Imaging Equipment.	Supplements ACR-NEMA commands with SPI commands for Imaging Equipment.	3
Presentation		Assembles/disassembles SPI messages. Converts Data Object formats (if required). Compresses/decompresses data (not implemented in the present SPI).	Uses ACR-NEMA message and data set standards. Supplements with clarifications and capabilities.	4
Session	Communication Interconnect	Establishes and maintains connection.		5
Transport	Communication Interconnect	Does name to address conversion. Provides reliable end-to-end connections using multiple virtual channels. Guarantees message delivery: accurately, on time, without duplication. Assembles/disassembles packets/messages for other levels. Controls flow of messages.	Fills void in the ACR-NEMA Standard.	5
Network	N/A	Inter-network functions for processing and routing are not included in SPI. These are pass-through services on SPI.	Not defined in SPI. IE to NIE connection.	N/A
Data link	Communication Interconnect	Makes efficient use of physical links (channels). Does link management. Does frame definition and creation. Provides flow control and error handling across physical links.		5
Physical	Communication Interconnect	Specifies mechanical, electrical, functional and procedural rules for access to physical communications channels.		5

CHAPTER 4

REQUIREMENTS

4.1 APPLICATION SERVICES

The term "Service" is used to designate a set of functional operations that are realized through commands, parameters, and procedures.

Application functions provided by SPI are:

- To send Data Objects to and receive Data Objects from a designated IE.
- To query an IE to determine whether specified data exists there or elsewhere.
- To request an IE to send specified Data Objects to the requestor.
- To request an IE to move specified Data Objects to a third IE.
- To exchange unformatted information between IEs.
- To issue or respond to a cancellation request relating to a previously required action to a query, or a send or move of a Data Object.
- To make a Data Object known to the IMS, or remove it from IMS knowledge (see 4.1.1).
- To change the state of a Data Object.
- To test and verify the end-to-end connections between IEs.
- To disseminate and monitor system configuration and identification data.

These functions can be considered to be segmented into functional modules termed "Application Services". The set of "Application Commands" is used to provide the services. The following Application Services are distinguished:

IMS - Image Management Service
 PBS - Public Storage Service
 PRS - Private Service
 EXS - Export Service
 CFS - Configuration Service

Not all services need be implemented in each IE, but one IE may have several services of the same or different type. In a PACS there is at most one implementation of the IMS, while there may be several nodes having temporary or permanent Data Object storage capability.

4.1.1 Image Management Service (IMS)

IMS consists of the following functions:

- To maintain a unique directory of Data Objects in one SPI PACS from the moment they are submitted until they are removed.

- To record information about the location of accessible images and other Data Objects. The storage of these IMS-known Data Objects is independent from the management function.
- For retrieval purposes, to query and interrogate the database that contains information about Data Objects and other information.
- To initiate transfer to each destination that requests information.
- To keep track of each Data Object's state.

4.1.2 Public Storage Service (PBS)

PBS provides temporary and/or permanent storage of IMS-known Data Objects. Data Objects stored on PBS are accessible to other IEs. An IE with PBS can send IMS-known Data Objects to another IE on request. PBS shall not delete an IMS-known Data Object without permission from IMS.

4.1.3 Private Service (PRS)

PRS provides the capability to receive/send IMS-unknown Data Objects between IEs. If PRS stores Data Objects, it may provide temporary and/or permanent storage. How this is done is defined in the implementation of PRS in the IE; there are no standard commands for PRS.

4.1.4 Export Service (EXS)

EXS is the capability to create copies of Data Objects on off-line media. EXS provides the creation of hardcopy and/or removable data exchange volumes.

4.1.5 Configuration Service (CFS)

CFS manages the information exchange between IEs based on the service, storage and command capabilities of each IE.

4.2 DATA OBJECTS

4.2.1 Classification of Data Objects

Each Data Object to be transferred between SPI IEs shall satisfy the format requirements of the ACR-NEMA standard for image exchange, (be "ACR-NEMA-conforming"), and shall satisfy the requirements of SPI (be "SPI-conforming").

4.2.2 Location of Data Objects

The IMS retains the logical addresses of IEs at which each IMS-known Data Object is stored.

4.2.3 Identification of Data Objects

Each SPI Data Object shall be uniquely identifiable by a fixed group of its elements; see Document 4.

4.2.4 State of a Data Object

Each Data Object is either IMS-known or IMS-unknown. IMS-known Data Objects have additional "state" information. The "state" information consists of state labels that name the categories, and a string value for each label. There are two classes of state labels: SPI-defined labels and non-SPI (private) labels. The SPI-defined labels are "ARCHIVED" and "REPORTED". Maintenance of the values of state labels is only performed by IMS. The IMS is not required to inform each IE that holds a copy of a Data Object about changes in state for the Data Object.

4.2.5 Data Object Type

Each Data Object shall have an element indicating the Data Object type. The following types shall be recognized:

Image	Private Image
Graphics	Private Graphics
Text	Private Text
Other	

These are defined as "Data Set Types" by ACR-NEMA. Document 4 defines several SPI Data Object types that are ACR-NEMA type "Other".

4.2.6 Protection

SPI provides no explicit means for read or write protection of Data Objects, but IEs are free to have such means.

4.2.7 Commands

SPI provides Application Commands to implement and utilize the services defined in 4.1.

If Application Services are requested for a data set that is not SPI-conforming, the results of the request are unspecified.

All SPI-conforming commands shall be recognized and responded to.

Each IE has the right to refuse acceptance of any commands for any reason. If an IE accepts a command, it must obey the ACR-NEMA and SPI rules governing the command.

4.3 TRANSMISSION PROTOCOL

The ACR-NEMA transmission protocol is defined in the ACR-NEMA Standard. See also the discussion in Document 5.

4.4 PHYSICAL LINK AND ELECTRICAL CHARACTERISTICS

The ACR-NEMA physical link is defined in the ACR-NEMA Standard. See also the discussion in Document 5.

4.5 OFF-LINE MEDIA FORMATS

SPI explicitly standardizes the following storage media:

- Optical Disk, 12 inch diameter.
- Optical Disk, 5.25 inch diameter.

The media formats are defined in Document 6.

It shall be possible to interchange the same kind of storage media between different IEs independent of the computer or operating system used.

CHAPTER 5

DATA INTEGRITY

At the application level, the SPI shall be capable of maintaining correct sequencing and integrity of transmitted data in an electrically noisy environment.

At the transport level, the SPI shall guarantee message integrity. For the network-independent part of the SPI, the error detection scheme shall cover noise pickup in both directions and all data transmitted (control, status, user data).

Failure of any part of an IE attached to the Network Interface Equipment shall not cause failure of the entire system or of any function except those in which the failed IE is directly involved.

CHAPTER 6

SERVICEABILITY

The SPI implementation in a system shall be capable of supporting testing and fault diagnosis. This shall include traffic monitoring and on-line or off-line local and remote loopback test facilities.

Other functional requirements are:

- Configuration/reconfiguration of addresses.
- Customization, system reconfiguration.
- Introduction of new releases.

CHAPTER 7

VERIFICATION

7.1 SCOPE OF VERIFICATION

Verification is a rigorous and controlled test against requirements. The intent of verification testing for SPI is to determine whether the hardware and software are complete and correct and in compliance with the SPI.

This completeness and correctness shall be demonstrated in a physical set-up that is representative of the types of IEs used in a PACS.

7.2 TEST PROCEDURE

The test procedure shall be described by a test plan that shall be audited and accepted by an authorized team.

The test plan shall describe the following items:

- Test designation, referring to the particular function or hardware/software element under test.
- Purpose of the test.
- Test bed configuration.
- Specification, option or feature being tested.
- Range of interface parameters being tested.
- Method of test, i.e., the sequence of operations.
- Inputs required for the test.
- Outputs expected for the test.
- Estimated time for the test.
- Criteria for successful testing: acceptance of test results.

The test procedure shall be integrated into the configuration control system for the SPI.

7.3 TEST CONFIGURATION

The test configuration shall consist of a typical system setup of IEs integrated through a network with the necessary SPI adapters and NIEs.

The test configuration shall permit all Application Services to be exercised.

7.4 PROGRAM OF TESTS

The following tests and services shall be possible:

Tests

1. "Power-on" self test (SPI plug disconnected)
2. Network functionality test.
3. IE/NIE integrity test (NIE disconnected from the network).
4. Functional system communication test (communication with each other node).
5. Functional system functions test (exercise all functions).
6. System performance test.

Services

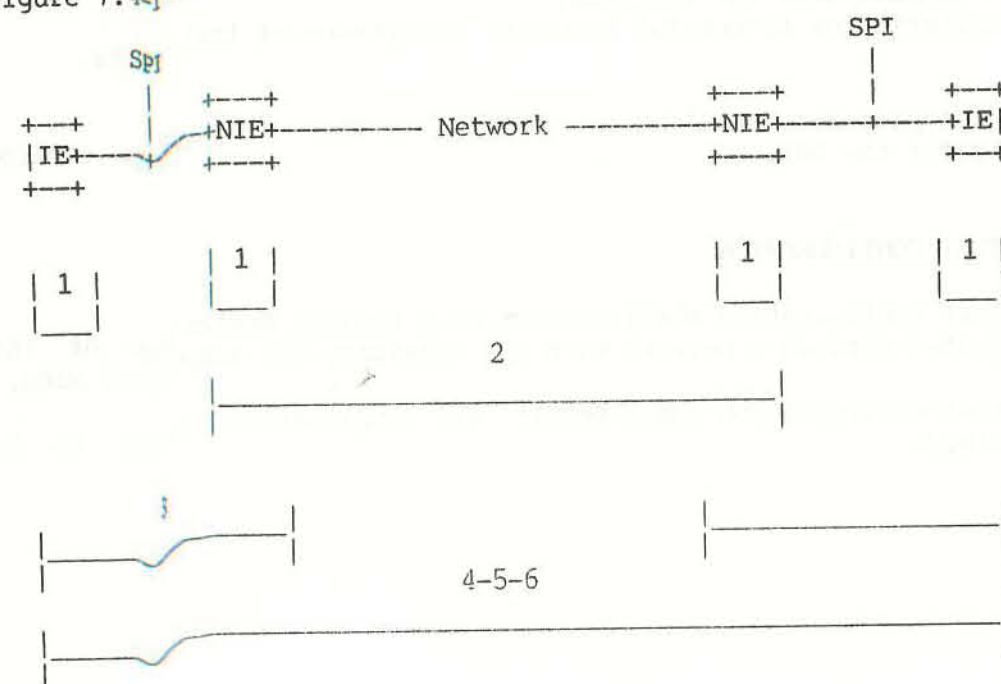
1. Set configuration.
2. On/Off-line announcement.

Notes

- Tests 1 and 2 are non-SPI-related tests.
- Tests 3 to 6 are SPI-related tests.
- The services are SPI-related and are necessary for the IE to become operational.
- There are no special extra commands in SPI necessary for the above-described tests; all test can be performed by using the already-defined SPI and ACR-NEMA commands.

The various tests cover the system elements as indicated in Figure 7.4-1, below.

Figure 7.4-1 Test Coverage



INDEX

- A -
 - Application Command 11*, 17
 - Application Service 15*, 21
- C -
 - Configuration Service 15-16
- D -
 - Data Object 16*
 - classification 2*, 16
 - identification 17
 - IMS-known 16*, 16-17
 - IMS-unknown 16-17
 - location 16
 - State 16, 17*
 - State labels 17
 - structure 11
 - Type 17*
 - Data Set Type 17
- E -
 - Export Service 15-16
- H -
 - High Level Function 11
- I -
 - Image Management Service 15*, 17
 - Imaging Equipment:
 - main functions 7
 - non-SPI-conforming 3
 - SPI-conforming 3
- L -
 - logical address 7
- M -
 - media format 18
 - message 11
- N -
 - name, logical = logical address
 - Network Interface Equipment 7-10
- O -
 - off-line media 18
 - optical disk 18
 - OSI Model 11
 - vs SPI 12-13*, 14
- P -
 - physical link 11, 18*
 - Private Service 15-16
 - Public Storage Service 15-16
- S -
 - SPI:
 - = Standard Product Interconnect adapter 9-10, 21
 - environment 7*
 - goals 2*, 5
 - requirements 3*
 - scope of compatibility 5-10*
 - Standard Product Interconnect 2*
 - State:
 - "ARCHIVED" 17
 - information 17*
 - private 17*
 - "REPORTED" 17
 - SPI-defined 17
- T -
 - Type, Data Object:
 - Graphics 17
 - Image 17
 - Other 17
 - Private Graphics 17
 - Private Image 17
 - Private Text 17
 - Text 17



STANDARD PRODUCT INTERCONNECT

for

COMPATIBILITY OF DIGITAL IMAGING

NOT TO BE DISCLOSED

SPI Document 3

Application Services

Document Version No. 1 Date: 1987-12-01
Document Edition No. 1 Date: 1987-12-01

This document is part of the SPI Release defined in the current "SPI Valid Documents List".

The document consists of 67 pages excluding this cover page.

Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved even in the event of the grant of a patent or the registration of a utility model or design.

CONTENTS

1.0	INTRODUCTION	3
1.1	OBJECTIVES	3
1.2	SCOPE OF DOCUMENTS	3
1.3	APPLICABLE DOCUMENTS	3
1.4	OTHER CONSIDERATIONS	4
2.0	SCOPE OF APPLICATION SERVICES	5
2.1	INTRODUCTION	5
2.2	DATA OBJECTS	6
2.2.1	Classification	6
2.2.2	Types	7
2.2.3	Identification	8
2.2.4	Location	9
2.2.5	State	9
2.3	APPLICATION SERVICES OF IMAGING EQUIPMENT	10
2.4	INTRODUCTION TO HIGH LEVEL FUNCTIONS	11
2.4.1	Purpose Of The Chosen Style Of Definition	11
2.4.2	Logical Model Of An Imaging Equipment	11
2.4.3	Conforming And Non-Conforming Situations	12
2.4.4	Acceptance Of Commands	12
2.4.5	Purpose Of Each High Level Function	13
3.0	HIGH LEVEL FUNCTIONS	16
3.1	ARGUMENT CONVENTIONS	16
3.2	EXPLANATION OF THE DESCRIPTION FORM	16
3.3	FUNCTION DESCRIPTIONS	18
3.3.1	SUBMIT	18
3.3.2	CHANGESTATE	19
3.3.3	FORGET	20
3.3.4	COPY	21
3.3.5	QUERY	23
3.3.6	GROUPCOPY	26
3.3.7	CONFIGURE	27
4.0	TRANSLATION OF HIGH LEVEL FUNCTIONS	28
4.1	MESSAGE DEFINED COMMUNICATIONS	28
4.2	TRANSLATION TABLE	29
4.2.1	Explanation of the Description Form	29
4.3	TRANSLATION OF GROUPCOPY	34

5.0	APPLICATION COMMANDS	36
5.1	GENERAL ASPECTS	36
5.2	APPLICATION COMMAND DEFINITIONS	38
5.2.1	READY/SPI	38
5.2.2	SEND/SPI	40
5.2.3	MOVE/SPI	42
5.2.4	SUBMIT/SPI	44
5.2.5	CHANGESTATE/SPI	45
5.2.6	DESUBMIT/SPI	46
5.2.7	FORGET/SPI	48
5.2.8	QUERY/SPI	49
5.2.9	COPY/SPI	51
5.2.10	CONFIGURE/SPI	52
5.2.11	GCBEGIN/SPI	54
5.2.12	GCEND/SPI	55
5.3	HANDLING OF ACR-NEMA COMMANDS	56
5.4	MAPPING APPLICATION COMMANDS TO ACR-NEMA	56
5.5	APPLICATION COMMANDS / APPLICATION SERVICES	57
6.0	APPLICATION SERVICES	58
6.1	INTRODUCTION	58
6.2	IMAGE MANAGEMENT SERVICE	58
6.3	PUBLIC STORAGE SERVICE	60
6.3.1	Remarks on Commands to PBS	61
6.4	PRIVATE SERVICE	62
6.4.1	Remarks on Commands to PRS	62
6.5	EXPORT SERVICE	63
6.6	CONFIGURATION SERVICE	64
INDEX		65

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVES

One of the major functions in a Picture Archiving and Communication System (PACS) is the exchange of images between various PACS components. This document addresses the flow of images and other Data Objects (DOs) in a PACS system and defines the related Data Object management functions. This functionality is provided by Application Commands. These are the commands between one application in one Imaging Equipment (IE) and another application in another IE.

The ACR-NEMA Standard specifies a standard image exchange format. The SPI extends the ACR-NEMA Standard for data and commands to improve PACS component performance and to satisfy such system requirements as system configuration and diagnostic testing.

The ACR-NEMA Standard is incorporated as a proper subset of the SPI-defined data and commands. See Document 2 for a discussion of basic SPI and Data Object requirements with respect to the ACR-NEMA Standard.

1.2 SCOPE OF DOCUMENTS

Document 3 deals with Data Objects, the transfer of Data Objects between different IEs and with the management of Data Objects.

A glossary of terms and a description of basic definitions of SPI notation are given in Document 1. Document 2 gives an overview of the concepts and global objectives of SPI. Documents 4, 5 and 6 contain detailed descriptions of Data Object Formats, communications standards and off-line media data formats, respectively.

1.3 APPLICABLE DOCUMENTS

The ACR-NEMA Standard is defined in the following document:
 "Digital Imaging and Communications Standard", ACR-NEMA Standards Publication No. 300-1985.

1.4 OTHER CONSIDERATIONS

Double quotes ("...") are normally used in the SPI documents; single quotes ('...') are used for special purposes in syntax definitions and inside text that is already quoted.

SPI ignores upper/lower case differences in ASCII strings used in command and Data Element text. The convention in other text is to capitalize commands and specific terminology for emphasis. Terms are not capitalized in general use. Thus, for example "Data Object" and "Data Element" are capitalized, while "object" and "element" are not. A special case, "data set", remains uncapitalized (except in tables and special lists), in conformance with ACR-NEMA usage.

Dates that are not part of SPI-defined syntax are written in ISO-format, e.g., 24 September 1987 becomes 1987-09-24.

In the text of this document, the convention is adopted that a hexadecimal number is designated by appending the letter "H" to the hexadecimal digit string.

CHAPTER 2

SCOPE OF APPLICATION SERVICES

2.1 INTRODUCTION

Application Services involve Data Objects and actions on/with Data Objects. These include the transfer of Data Objects between IEs and the management of Data Objects by the Image Management Service (IMS).

The operations allowed on a Data Object by SPI depend on whether or not the Data Object has been registered with the Image Management Service. Registered Data Objects are referred to as "IMS-known". SPI imposes rules regarding the handling of "IMS-known" Data Objects. These rules guarantee the integrity and consistency of all such Data Objects and are required for SPI conformance.

When a Data Object is not registered with IMS, it is referred to as "IMS-unknown". Such Data Objects are unregulated by the SPI within an Imaging Equipment, since they are considered local, i.e., they are "private property" of the IE.

All communication in the SPI is defined by messages based on the ACR-NEMA standard.

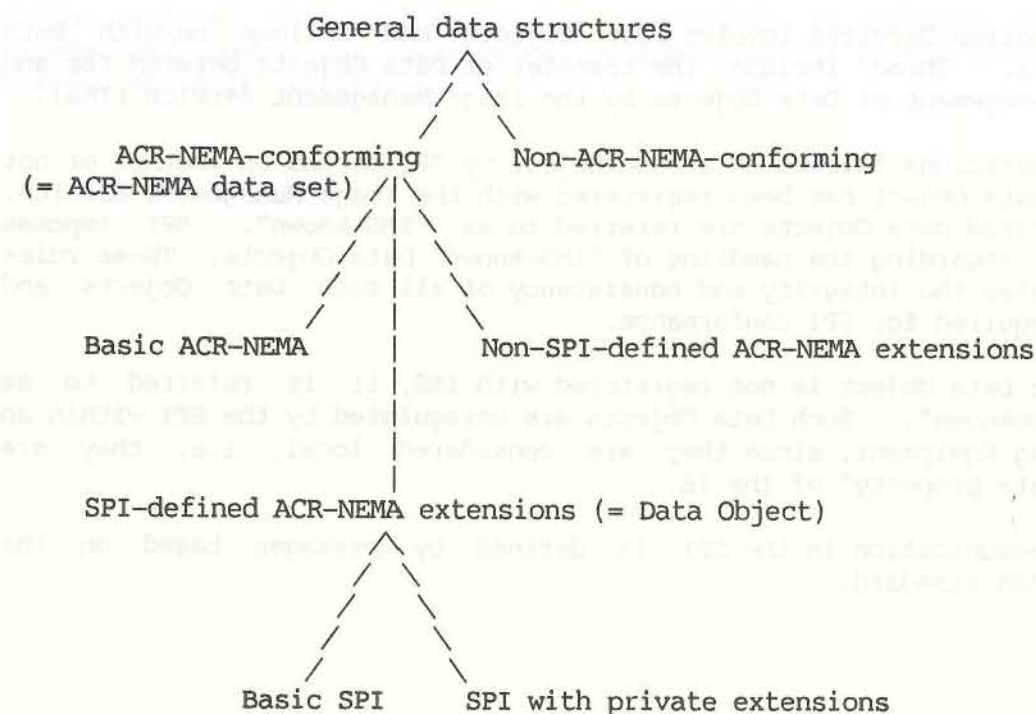
2.2 DATA OBJECTS

Within SPI, only Data Objects conforming to certain rules are allowed. The basic consideration for implementing an SPI-conforming IE is attention to the requirements of Data Objects.

2.2.1 Classification

General data structures used by IEs can be classified as follows in Figure 2.2-1.

Figure 2.2-1 Data Structures Classification



2.2.2 Types

A data structure is a Data Object if it conforms to the following rules:

- It conforms with the ACR-NEMA standard.
- It carries one of the following Data Set Types defined within ACR-NEMA:

Image	Private Image
Graphics	Private Graphics
Text	Private Text
Other	

Note:

An ACR-NEMA Data Set Type "Other" includes all additional Data Objects defined within SPI and all SPI-conforming manufacturers' private additions. Document 4 defines the Data Object Types and coding rules. Rules for private additions are also supplied.

- It carries the identification defined by SPI (UID).
- Data Object Description elements (DESC) for every Data Object Type shall be as defined by Document 4.
- Data Objects whose type is not defined within Document 4 shall contain a formal definition of their DESC Data Elements.

2.2.3 Identification

The ACR-NEMA Standard does not define a specific means of unique data set identification, although it does indicate certain possibilities for data sets of type "image". To assure integrity of operations in an SPI-based PACS, SPI requires that each Data Object can be uniquely distinguished from every different DO. Since the ACR-NEMA Standard does not assure this except via bitwise comparison, SPI defines an element for this purpose called the UID, or Unique Identifier. The UID is normally placed into a DO at the moment of its creation by an SPI-conforming IE. Non-SPI IEs cannot be expected to provide this UID; therefore, when an SPI IE receives a copy of a data set from a non-SPI IE, the SPI IE is responsible for adding a UID at that time. Similarly, when an SPI IE transmits a copy of a Data Object to a non-SPI IE, the SPI IE is responsible for removing the UID and other SPI-added elements from the transmitted copy. The purpose of removing these is to assure that when a data set is received from a non-SPI IE, and later returned to one, the data set returned is identical to the data set that was sent. In other words, the UID and other SPI-specific elements are only necessary, and only present, within the "SPI domain".

The following protocol is required for modification of Data Objects: When a Data Object is modified in any way, a new UID shall be created, since the new DO is different from the old DO. Instead of simply discarding the old UID, the IE making the change shall place a text string that explains the change into the lowest available even-numbered element in a set of elements reserved in each DO for recording the modification history (see Document 4). The old UID is placed into the following (odd-numbered) element. Thus, the entire modification history is kept with the DO. For system integrity, this procedure shall be followed whenever there is a UID element present in the DO, whether or not the DO is known to the IMS.

In SPI, two Data Objects that have identical UIDs are required to have identical contents. If an SPI-conforming IE changes the contents of a Data Object, whether it is "IMS-known" or not, it is required to change the UID. If an IMS-unknown copy of a Data Object is not changed, an SPI-conforming IE should not change the UID.

Note that, analogous to the ACR-NEMA element (0008H,0040H), a UID in a message often identifies a Data Object not present in the message. Remember, a UID is part of every DO.

The set of Data Elements that together constitute a given Data Object is divided by SPI into three disjoint subsets. The first subset is the UID element mentioned above. SPI also defines, for each Data Object Type, a set of elements called the Data Object DESCription (DESC). The third set is simply the remaining elements. The definitions of these sets may be found in Document 4. The union of UID and DESC sets is intended to cover the elements expected to be queried for Data Object selection; an SPI-conforming IMS is required to store these elements.

Data Objects of a type not defined in Document 4 shall include a free-formatted DESC element that specifies the list of (Group,Element) pairs that form the DESC subset.

2.2.4 Location

The place where a copy of a Data Object is stored is called its location. The Data Object location is specified by the logical address of the IE where the Data Object is stored. Copies of the same Data Object may be stored at more than one location.

2.2.5 State

Usage of an SPI-based PACS containing an IMS is facilitated by definition of certain "State" information for Data Objects, and the maintenance by IMS of this data. State information consists of State labels that name the categories, and an ASCII-string value for each label.

The "ARCHIVED" State is defined by SPI: it may have one of the values "YES" or "NO". An IE shall not change the "ARCHIVED" State to "YES", unless there is an IMS-known copy of the Data Object on permanent storage.

The "REPORTED" State is defined by SPI: it may have one of the values "YES" or "NO". SPI defines no specific rules regarding this State label.

"Private" (non-SPI) labels may be used. An SPI-conforming IMS shall provide at least 32 characters for storage of private states and their values in addition to the defined States and values.

The IMS will change the value of the State label when a change of State is reported to the IMS. Note that changes in State do not necessarily correspond to changes in a Data Object: if the Data Object is changed, then its UID shall change, and perhaps its State does, but changes in State can also occur without DO changes. For example, the "REPORTED" State is intended to reflect that a diagnostic report has been prepared for a DO: if the report text has been added to the DO, then both its Unique Identifier (UID) and its "REPORTED" State should change. However, if the report is external to the PACS, the report causes no change to the DO at all, but the fact that the diagnosis has been performed may still be reflected by a change of the "REPORTED" State from "NO" to "YES" in the IMS.

An IMS is not required to inform any IE that holds a copy of a Data Object about changes in State for the DO. Therefore, the knowledge at a given IE about the State of a DO it holds may be obsolete.

2.3 APPLICATION SERVICES OF IMAGING EQUIPMENT

An Application Service is a set of capabilities that is optionally provided by an IE to process a defined set of incoming commands.

Document 2 defines the following Application Services:

Service	Definition
PBS	Public Storage Service. PBS is the capability of storing "IMS-known" Data Objects. PBS provides temporary and/or permanent storage.
PRS	Private Service. PRS is the capability of receiving/sending IMS-unknown Data Objects. If PRS stores Data Objects, PRS may provide temporary and/or permanent storage.
EXS	Export Service. EXS is the capability of creating copies of Data Objects for use outside the PACS. EXS provides the creation of hard copies and/or off-line data exchange volumes.
IMS	Image Management Service. IMS keeps records of "IMS-known" Data Objects. IMS responds to QUERY REQUESTS. IMS is resident on at most one IE in one SPI network.
CFS	Configuration Service. CFS manages the information exchange between IEs about their capabilities.

2.4 INTRODUCTION TO HIGH LEVEL FUNCTIONS

2.4.1 Purpose of the Chosen Style of Definition

To implement a full, practical PACS, it is necessary to add to the commands provided by the ACR-NEMA Standard. Thus, SPI extends ACR-NEMA to make such a PACS possible. In providing these extensions, SPI encourages the use of the ACR-NEMA Standard, both by incorporating it as a subset, and by using its means to realize the needed extensions.

SPI facilities are defined in terms of High Level Functions (HLFs). These HLFs subsume and extend the basic ACR-NEMA set. The motivation for the introduction of HLFs, rather than just extensions of the commands given by ACR-NEMA, is to provide a concise, abstract definition of SPI capabilities that is relatively isolated both from the details of the commands provided in the ACR-NEMA Standard, and from possible future changes to them. These High Level Functions are defined in Chapter 3; the Application Commands that implement them are given in subsequent chapters.

2.4.2 Logical Model of an Imaging Equipment

An IE in an SPI PACS can be thought of as containing several conceptual layers of software: First, there is an IE-dependent layer that interacts with the person operating the IE in accordance with the "personality" and user interface of that equipment. Having determined the wishes of its user, this layer interacts with the second layer. The interface between the first and second layers is the set of High Level Functions. The second layer translates these High Level Functions into commands, using shadow groups and/or other ACR-NEMA extensions as appropriate. It is emphasized that these layers are conceptual; an SPI-conforming IE may implement its functions in any manner that provides the prescribed semantics of the HLFs in Application Commands.

2.4.3 Conforming and Non-Conforming Situations

When Application Services are requested for an ACR-NEMA data set that is non-SPI-conforming, the results of the request are unspecified, except in a few cases for which a particular result is defined. When basic (unextended) ACR-NEMA functions are requested, the results shall conform to the ACR-NEMA Standard. The SPI response to Service requests containing non-SPI extensions to ACR-NEMA commands is unspecified.

The reason for defining SPI behavior as "unspecified" in certain situations is to make its definition open-ended, and to avoid requiring that SPI-conforming IEs shall implement checking or enforcement functions in their implementations. The open-endedness occurs because manufacturer/user-defined extensions to ACR-NEMA and SPI are allowed, since presently unused "features" are not defined to be error cases. Since an SPI-conforming IE has no responsibility for what happens if it should be given a non-conforming command or Data Object, there are no implementation or run-time requirements associated with enforcing SPI rules when dealing with "out-of-spec" or "foreign" inputs.

2.4.4 Acceptance of Commands

To permit a wide range of choices of architectures, systems, and behavior in a PACS based upon SPI, only a relatively small number of restrictive rules have been included. A fundamental premise in SPI is that each modality is allowed to retain as much autonomy as possible, consistent with overall system integrity requirements. A consequence of this autonomy is that each IE has the right to refuse any request from another IE based upon whatever criteria it chooses. If it accepts the request, it shall conform to the SPI definition of how the request is to be performed. If it refuses, it shall do so in the manner prescribed by SPI.

2.4.5 Purpose of Each High Level Function

The High Level Functions are:

Function	Purpose
SUBMIT	Make a Data Object known to IMS.
CHANGESTATE	Change the State information of an "IMS-known" Data Object.
FORGET	Erase IMS knowledge of one or more copies of a Data Object.
COPY	Transmit a Data Object from one IE to another.
GROUPCOPY	Transmit a collection of Data Objects to a designated IE.
QUERY	Obtain information about Data Objects or other information known to the addressed IE.
CONFIGURE	Exchange capability information with one or more IEs.

SUBMIT and FORGET

In the following discussion of SUBMIT and FORGET, it is assumed there is an Image Management Service (IMS) in the PACS; otherwise, SUBMIT and FORGET are not supported.

One or more copies of a given Data Object may be stored in the IEs of a PACS. When at least one of the copies is registered by the IMS, the Data Object is said to be "IMS-known". Furthermore, each copy of a DO is "IMS-known" or not, individually, depending upon whether or not the IMS is aware of its existence: each one registered with the IMS is said to be "IMS-known", and each one not registered is said to be IMS-unknown. Thus, when a Data Object is unknown to IMS, all of its copies are IMS-unknown. When IMS has knowledge of a copy of a Data Object, the DO and that copy are "IMS-known"; other copies, if any, may or may not be.

The SUBMIT and FORGET Functions are a complementary pair. SUBMIT is used by an IE to register a copy of a Data Object it holds with IMS. When IMS "registers" a copy, it stores identifying and descriptive information for the DO and the location (IE) at which the copy is stored. Since SPI defines a means for unique identification of SPI Data Objects, IMS can recognize when a DO being submitted is already registered: if the DO is not already known by IMS, then IMS shall add the identifying and descriptive information for the DO, and its location, to its database. However, when the IMS is asked to register another copy, it only needs to append the new location to its database.

An IE that holds an "IMS-known" copy of a Data Object shall not delete or modify that copy as long as it is "IMS-known". Thus, the concept of "IMS-known" involves cooperation between the IEs that store such Data Objects and the IMS.

FORGET is used by an IE to request erasure of knowledge of one or more copies of a Data Object by IMS, subject to agreement by the IE where each such copy is stored. When FORGET causes IMS to erase its knowledge of the last "IMS-known" copy of a DO, the DO becomes IMS-unknown.

When IMS has knowledge of one or more copies of a Data Object, it is capable of returning information about the DO in response to a QUERY REQUEST issued by an IE; otherwise, it is not.

CHANGESTATE

The "State" of a Data Object provides operational information, such as whether or not the DO has been archived, or whether a diagnosis has been reported.

The CHANGESTATE Function, supported only when an IMS is available in the PACS, allows the modification of the IMS-maintained State information for an "IMS-known" Data Object. Because the State information for a DO is only held by IMS, the knowledge of the State of a DO at a particular IE may be outdated. The QUERY Function may be used to obtain the current State value from the IMS.

COPY

The COPY Function is an abstraction of the concept of transmitting a copy of a Data Object from one location in a PACS to another. It includes the capability provided by the SEND, GET, and MOVE commands of ACR-NEMA, and adds further SPI functionality. In particular, the IE performing the COPY Function can specify the source and destination for the transfer and whether the new copy is to be made known to the IMS. Additionally, the choice of the IE to be used as the source for the copy can be left to IMS.

Each successful invocation of the COPY Function creates an additional copy of the Data Object being transmitted; the copied Data Object remains in its original place.

When there is no IMS available in the PACS, those parts of the COPY Function that do not require IMS are supported; the remainder are not.

GROUPCOPY

Just as ACR-NEMA restricts its messages to contain only one data set, each performance of COPY is limited to transfer of a single Data Object. However, in a PACS, actual operations sometimes require transmission of a group of related Data Objects. Examples include archiving an entire set of CT slices, and directing a format and a group of image Data Objects to a digital hard copy device. The GROUPCOPY Function allows such operations to occur in a single session: GROUPCOPY transmits a sequence of one or more Data Objects, possibly from more than one location, to a single destination.

QUERY

The QUERY Function extends the ACR-NEMA FIND command in terms of richness of inquiry and in specification of the contents and other properties of the responses desired. Perhaps the most typical use of QUERY is to obtain information from IMS; however, for generality, QUERY can be used with any IE that will accept it.

CONFIGURE

Normally, an SPI-based PACS will involve multiple IEs. The CONFIGURE Function provides a means for an IE to announce its set of services and capabilities to one or more IEs in the configuration, and to receive service and capability information back from them in return. Two types of response may be requested by the originator from the IEs it addresses: a "COARSE" response providing information sufficient for the majority of purposes, such as gross capabilities, and their current on/off-line status, and a "FINE" response, giving details that are relatively static but that require a relatively large description. The "FINE" request can be addressed to any one IE at a time, whereas the "COARSE" request can be sent to more than one IE, or to all IEs. An example of where a "FINE" response is needed from an IE is to define exactly the type of QUERY it will accept. Such a definition (see Document 4) consists of a list of all Data Elements it retains and those relational and other operators supported.

CHAPTER 3

HIGH LEVEL FUNCTIONS

The following High Level Function (HLF) definitions describe operations allowed in an SPI-conforming PACS. The SPI-conforming implementation of these functions in terms of Application Commands is given in later chapters. The result of a non-conforming invocation of a HLF is unspecified.

3.1 ARGUMENT CONVENTIONS

In the descriptions below, the function argument called <TEMPLATE> may be a partial specification of a Data Object (in which one or more elements are specified by a pattern, or are not specified at all) designating a set of zero or more Data Objects, depending on the number of matches which occur.

The <LOCATION>, <SOURCE>, and <DESTINATION> arguments consist of an IE logical address. The designation of a "Service" subaddress within that IE is allowed as a separate argument when necessary.

Several arguments, not explicitly shown in the argument lists, shall be added by the IE when the HLF is translated to the proper sequence of Application Commands (refer to Chapter 4).

3.2 EXPLANATION OF THE DESCRIPTION FORM

The description form for each High Level Function consists of five parts: Function Format, Effect, Actions, Constraints, and Remarks.

The "Function Format" shows the function name, and required and optional input arguments. The results of performing the function are returned to the issuer in an implementation-specific manner.

"Effect" is a brief statement of the major result of the function.

"Actions" describes the semantics of the function: actions are performed sequentially in the order listed; if an action results in the occurrence of an error response, the action sequence terminates at that point. Although not always explicitly shown in the Action list, when an action requires communication with an IE not present in the PACS or not currently on line, an error response occurs.

One Action class deserves special mention: the set that manipulates the so-called "IMS-known" attribute. The "IMS-known" attribute is a conceptual two-valued entity that provides an IE-internal record for each Data Object held by that IE. When its value is "YES", it signifies that that copy is presently registered with IMS; when its value is "NO", the copy is not registered. The descriptions below do not repeat the fact that this is a conceptual attribute; however, the means actually used by a given IE to "remember" which Data Objects are known to the IMS are outside the scope of SPI.

"Constraints" are restrictions that shall be obeyed by each conforming IE.

"Remarks" adds information to the semantic definition when there is no other appropriate way to provide it in the description. It also provides informal guidance to interpretation.

3.3 FUNCTION DESCRIPTIONS

3.3.1 SUBMIT

SUBMIT (UID, DESC [, OPTIONAL-ELEMENTS])

A. Effect:

Make an SPI-conforming Data Object known to IMS.

B. Actions:

1. If IMS does not accept the SUBMIT request, an error response occurs.
2. The IMS registers the existence of the submitted copy of the Data Object, retaining the <UID> and all provided <DESC> elements, any elements it chooses from the group <OPTIONAL-ELEMENTS> (if present), and the location where the copy is stored.
3. The "IMS-known" attribute of the SUBMITTED Data Object is set to "YES" at the IE that originated the SUBMIT.

C. Constraints:

1. An IE shall not perform the SUBMIT Function if it does not hold a copy of the designated Data Object.
2. An IE shall not perform the SUBMIT Function for a copy of a Data Object whose "IMS-known" attribute is "YES" at that IE.
3. An IE shall not change or delete a Data Object while it is IMS-known.

D. Remarks:

1. The purpose of the SUBMIT Function is to cause IMS to record information concerning a) a "new" Data Object (one having a UID not previously known by IMS); or b) an additional copy of a Data Object (one whose UID is already IMS-known). Making a (copy of a) Data Object known to IMS results in IMS having the ability to answer inquiries about the DO, to perform other functions that partially or fully specify the Data Object, and to provide the State of the DO and information about locations at which copies of the Data Object are stored.
2. To make a modified version of an IMS-known Data Object, an IE shall create an IMS-unknown copy and modify that.
3. When an IE wishes to delete an IMS-known copy of a Data Object from its own storage, it shall first make the copy IMS-unknown via the FORGET Function.

3.3.2 CHANGESTATE

CHANGESTATE (UID, STATE)

A. Effect:

Record, within IMS, a <STATE> value for an IMS-known Data Object.

B. Actions:

1. If IMS does not accept the CHANGESTATE request, an error response occurs.
2. If the DO specified by <UID> is not known to IMS, an error response occurs.
3. If the requested state change violates an SPI or private rule implemented by IMS, an error response occurs.
4. The IMS registers the state information provided by <STATE> for the designated DO.

C. Constraints:

None.

D. Remarks:

1. An IE may perform the CHANGESTATE Function whether or not the IE holds a copy of the designated DO.
2. An IE may perform the CHANGESTATE Function for a copy of a DO whose IMS-known attribute is "YES" or "NO" at that IE; however, at least one copy of the DO must be IMS-known somewhere.
3. The form for <STATE> is <State label> '=' <value>. If the value is the null string, it shall be interpreted as a request to delete the label and its value from the record kept by IMS.

3.3.3 FORGET

FORGET (UID, LOCATION, FORGET-LAST-COPY)

A. Effect:

Remove one or more copies of a Data Object from IMS knowledge.

B. Actions:

1. If IMS does not accept the FORGET request, an error response occurs.
2. The <LOCATION> argument may indicate a) one or more IEs; or b) "ALL" IEs. The IMS creates a candidate list of locations where IMS-known copies at IEs specified by <LOCATION> exist. If this candidate list is empty because the Data Object is unknown, or because it is not known to exist at any of the specified locations, an error response occurs.
3. If the candidate list contains all locations at which the DO is known by IMS, and if the value of <FORGET-LAST-COPY> is not "YES", IMS removes an arbitrary entry from the list. If the resulting list is empty, an error response occurs.
4. For each entry on the list, in unspecified order, the IMS sends a FORGET request to the IE holding the Data Object. If the IE accepts the FORGET request, the IE shall set its "IMS-known" attribute of the Data Object to "NO", and the IMS shall erase its own knowledge of that Data Object for that location. If the IE does not accept the FORGET request, the "IMS-known" attribute at the IE remains set to "YES", and the IMS does not erase its record of that DO for that location.
5. The IMS returns to the IE that originated the FORGET Function the list of IEs where the Data Object was located, an indication of those that were "forgotten", and the reason for each that was not.

C. Constraints:

If an IE specifies the value "ALL" for <LOCATION>, it shall also specify the value "YES" for <FORGET-LAST-COPY>.

D. Remarks:

1. Whether an IE retains or deletes a Data Object is strictly under the control of that IE when the DO is not IMS-known.
2. An IE that considers itself to be an "archive" can be expected to refuse FORGET requests.

3.3.4 COPY

COPY (UID, SOURCE, DESTINATION, CLASS, IMS-FLAG)

A. Effect:

A copy of a Data Object is transmitted from one IE to another.

B. Actions:

1. The <SOURCE> argument may indicate either a) a specified location from which the copy is to be made; or b) that the source of the copy is to be chosen by IMS. In case a, if the Data Object designated by <UID> does not exist at the specified <SOURCE>, an error response occurs. In case b, if IMS does not accept the COPY request, or if the designated DO is not known by IMS, an error response occurs.
2. If the <DESTINATION> does not accept the proposed transmission of the copy, an error response occurs (see Remark 1).
3. A copy of the Data Object designated by <UID> is transmitted from the <SOURCE> to the <DESTINATION>.
4. If the <DESTINATION> does not accept the copy, an error response occurs.
5. If <IMS-FLAG> has the value "SUBM", the <DESTINATION> performs a SUBMIT Function for the Data Object upon its arrival; the result of the SUBMIT is returned to the originator of the COPY. If the result is "failure", additional information is returned indicating the reason for failure; the <DESTINATION> IE is expected to delete the (IMS-unknown) DO.

C. Constraints:

An IE shall not originate a COPY Function in which <DESTINATION> is the same as <SOURCE>.

D. Remarks:

1. There is an optional preliminary communication between two SPI-conforming IEs before the transmission of a "large" Data Object (such as an image) to establish agreement based on the size of the Data Object to be sent, the intended <CLASS>, and the intended "IMS-known/unknown" condition of the copy. This is provided for performance reasons: a destination can indicate refusal without requiring that the full transmission take place first. The apparent repetition in Actions 2 and 4 is due to the possibility that the indicated refusal could occur before or after the attempted transmission of the DO.

2. If <IMS-FLAG> has the value "NOTKNOWN", the destination IE can be expected to not SUBMIT the DO on arrival. However, the acceptance of a <IMS-FLAG> = "NOTKNOWN" copy of a Data Object by a destination IE does not preclude that IE from performing a SUBMIT for that copy of the DO.
3. The defined values of the <CLASS> argument are "PERM" (permanent), "TEMP" (temporary), "PRINT" (print), and "EXCH" (data exchange). Informally, permanent storage is considered permanent, and FORGET requests are therefore ordinarily refused. Temporary storage ordinarily does not refuse FORGET requests, and ordinarily will delete IMS-unknown DOs. Print indicates that hard copy services are wanted. Data-exchange is an indication that the copy is intended to be removed from the system and sent to another physical location.
4. When IMS is to choose the source from which the copy is to be made, its basis for the choice is unspecified.

****IMPLEMENTATION NOTE****

A third potential value of <IMS-FLAG>, "KNOWN", has been deferred for consideration until a later release of SPI. The meaning of "KNOWN" is that the transmitted copy is to be made known to IMS before it arrives at the destination. This would allow an IE to store IMS-known DOs without being required to have the capability of performing SUBMITs.

3.3.5 QUERY

QUERY (TARGET, [TARGET-SUBSET], TEMPLATE, LOCATION, MAX-RESPONSES, RETURN-INFO-FLAG, COUNT)

A. Effect:

Obtain information about Data Objects and other Data Elements known by the target IE.

B. Actions:

1. The query is transmitted to the <TARGET> node. The optional <TARGET-SUBSET> designates to which Service in the target the query is addressed: values may be "PBS", "PRS" or "IMS", where "PBS" means public or IMS-known storage, "PRS" means private or IMS-unknown storage and "IMS" means DOs known to that target's IMS. For each Data Object that satisfies the <TEMPLATE> criteria, the <TARGET> IE returns information controlled by the <RETURN-INFO-FLAG> and <COUNT> arguments. The <RETURN-INFO-FLAG> can have one of three values: a) include only the available information that corresponds to elements specified in the template; b) include information as in a, plus all available information for elements of DESC; and c) include all available information. In all cases, each match response contains a list of locations at which copies are known by the target to exist, and the "State" of the DO. The <LOCATION> argument may indicate a) a list of one or more IEs; b) "ALL" IEs; or c) "NONE" of the IEs. If <LOCATION> is a list, no match response will be returned that does not mention at least one of the IEs in the <LOCATION> list as a location. If the <COUNT> argument is "YES", then each match response returns the number of DOs that match the criteria; otherwise, the value of this count is not required in the responses.
2. The maximum number of match responses that the <TARGET> IE is permitted to return is given by the <MAX-RESPONSES> argument. If the number of match responses that otherwise would have been returned exceeds the value of <MAX-RESPONSES>, the final response from the <TARGET> IE includes a qualitative indication that there would have been more.

C. Constraints:

None.

D. Remarks:

1. Matches are not limited to Data Objects. In this case the <LOCATION> value in the request and response command is "NONE". This allows the possibility of obtaining non-DO-oriented information from an IE.
2. The <TEMPLATE> argument is used to specify criteria for selection of Data Objects from the set of Data Objects known by the <TARGET> IE, which normally is the IMS. It is typical that with the QUERY Function the <TEMPLATE> argument is only

partially specific in order to select more than one Data Object that satisfies the <TEMPLATE> criteria. QUERY can also be used to learn where copies of specific DOs are located.

3. The <TEMPLATE> argument is a predicate to be applied (in principle) to each Data Object known to be stored at locations of interest. If the value of the predicate is "TRUE", the Data Object is included in the response; otherwise, it is not included. Predicates are valid if and only if they can be generated from the following syntax:

```

<Predicate> ::= <Bool_Value>
              | <Paren_Bool_Value>

<Paren_Bool_Value> ::= '(' <Bool_Value> ')'

<Bool_Value> ::= <NEMA_Group_and_Element> <Relation> <Constant>
              | 'NOT' <Paren_Bool_Value>
              | <Paren_Bool_Value> 'AND' <Paren_Bool_Value>
              | <Paren_Bool_Value> 'OR' <Paren_Bool_Value>
              | 'STATE(' <State_label> ')' <Relation> <Constant>

<NEMA_Group_and_Element> ::= '[' <Group> ',' <Element> ']'

<Relation> ::= '=' | '<' | '>' | '>=' | '<' | '<='

<Constant> ::= <Integer> | 'NULL'
              | <ASCII_Numeric_String>
              | <ASCII_Text_String>

<State_label> ::= <ASCII_Text_String>

<Group> ::= <Even_Integer>
           | <Odd_Integer> ',' <Shadow_Owner_Code>

<Element> ::= <Integer>

<Shadow_Owner_Code> ::= <ASCII_Text_String>

```

<Integer>, <ASCII_Numeric_String>, and <ASCII_Text_String> are as defined by the ACR-NEMA Standard for the value representations BI, AN, and AT, respectively. <Even_Integer> represents an even <Integer> number, while <Odd_Integer> represents an odd <Integer> number. Within the two ASCII forms, the special characters '*' and '@' may be used with the meanings defined by the ACR-NEMA Standard. The value 'NULL' is used as a wildcard for integer values, having a meaning equivalent to '*' as used for strings.

Wildcard characters may be used for <State_label> and <Shadow_Owner_Code>, but an error response occurs if there is more than one match.

For ACR-NEMA shadow groups, a <Shadow_Owner_Code> specifies which "effective" <Element> is queried.

Note that parentheses designate the operands of each operator.

4. The "YES" value of the <COUNT> argument specifies a requirement that the <TARGET> IE is to include valid, non-zero values for the ACR-NEMA element (0000H,0850H) "Number of Matches" in its responses. The "NO" value is an indication that this value is not needed; the number and content of responses is not otherwise affected.
5. The interpretation of the <RETURN-INFO-FLAG> is equivalent to a template specification in which each of the additional elements implied by <RETURN-INFO-FLAG> had been explicitly coded with the '*' or 'NULL' wildcard value appropriate for its type.
6. A target not knowing the composition of the DESC set for a DO, but which has been asked to return DESC information, is allowed to return all available information instead of the specified subset.
7. The QUERY Function is not required to produce a sequence of responses that could have been generated from a single, coherent "snapshot" of the database.
8. The "STATUS = FINAL" QUERY RESPONSE is allowed to be separate from the last match response, rather than being included with it. (The "FINAL" status is any status other than "PENDING".) If it is separate, it is not counted with respect to <MAX-RESPONSES>.
9. The order of responses to a QUERY is unspecified; thus, if the sequence was stopped because the <MAX-RESPONSES> limit was reached, there is no defined way to obtain the "next" group of responses.
10. Examples of valid predicates are:

```

[8,20] <= "1984.07.@"
([10,1020] <= "1.4") OR ([10,1020] > "1.9")
([28,10] <> 256) AND ([28,11] <> 512)
[28,10] <> 'NULL'

```


3.3.6 GROUPOCOPY

GROUPOCOPY (DESTINATION, CLASS, IMS-FLAG, SOURCE-1, OBJ-1-UID,
 ...,
 SOURCE-n, OBJ-n-UID)

A. Effect:

Transmit a sequence of Data Objects to a specified destination as a single, bracketed session.

B. Actions:

1. For each Data Object to be transmitted, a COPY Function with arguments taken from the GROUPOCOPY counterparts is performed. If any COPY is unsuccessful, an error response occurs.

C. Constraints:

None.

D. Remarks:

1. The GROUPOCOPY Function is intended to allow an IE to transfer a group of Data Objects from one or more locations to a single destination in one session.
2. One use of GROUPOCOPY is for a digital hard copy print function described in 6.4 (Export Service).
3. The transmission of a Data Object from one SPI IE to another using SPI protocol is accomplished with an optional preliminary "handshake" communication, allowing the two IEs to agree or disagree on the transmission of the DO prior to an attempted actual transfer. GROUPOCOPY is accomplished with one additional preliminary communication to the receiver in which the planned number of DOs is specified.

3.3.7 CONFIGURE

CONFIGURE (TARGET, MY-CAPABILITIES, RESPONSES-FLAG [,CAPABILITY])

A. Effect:

Exchange capability information with one or more IEs.

B. Actions:

1. The <TARGET> argument may indicate a) a list of one or more IEs, or b) "ALL" IEs as the target(s) of the function. The issuing IE then transmits a "COARSE" capability list describing the functionality (<MY-CAPABILITIES>) it offers to the target IE(s).
2. The <RESPONSES-FLAG> argument may have one of three values: If <RESPONSES-FLAG> has the value "NONE", no capability list is required in the response from targets. The value "COARSE" requests addressed <TARGET> locations to respond with their own "COARSE" capability lists. The value "FINE" shall be used only with a <TARGET> argument that specifies a single IE; when "FINE" is specified, the optional argument <CAPABILITY> shall be present and shall specify a single capability of the target for which information is desired. The meaning of "FINE" is to request a response containing a fully-detailed description of that capability at the target.

C. Constraints:

None.

D. Remarks:

1. An SPI-conforming IE is expected to use the CONFIGURE Function to announce its entry to on-line status, and to announce its intention to go off-line when it does so in a controlled way.
2. When requested, the "COARSE" response from each target IE shall contain a capability list describing the "functionality" it presently offers on line to the originating IE. In the event that the <TARGET> argument specifies "ALL" IEs, the maximum number of IEs that might respond may not be predictable by the issuing IE; the minimum number may be zero.
3. Failures or other circumstances may result in an IE being unable to transmit a message indicating its intention to go off line before it actually does so.
4. The transmission of specific capabilities to some, but not all, IEs can be used when there are instances where different IEs should have different pictures of the capabilities of the issuer. An example is a testing environment, in which an IE is placed on line only with respect to a partner participating in the testing.
5. If the <TARGET> argument specifies more than one IE, then if one or more IEs in the list are not available, no error response occurs for those IEs.

CHAPTER 4

TRANSLATION
OF HIGH LEVEL FUNCTIONS

4.1 MESSAGE DEFINED COMMUNICATIONS

All SPI communication occurs via messages based on the ACR-NEMA Standard. A message consists of a command and an ACR-NEMA data set (which may be null).

A message that is sent by an IE will effect one or more subsequent messages. The protocol for Application Commands is based on a sequence of messages defined in this document. The simplest protocol is a command request from the initiator of the command to the receiver of the command; the receiver returns a command response to the initiator. This protocol is defined by two messages between two IEs. There are other protocols with more than two messages and more than two IEs.

The High Level Functions defined in Chapter 3 are implemented by Application Commands transmitted between IEs with such messages. Each Application Command addressed to an IE is defined to produce zero or more responses containing the "PENDING" Severity Status Code indicating that the command has been correctly received and that execution is in progress. Each command also produces exactly one response containing the status "FINAL" that indicates the completion of its execution. The "PENDING" response(s) and the "FINAL" response may all convey additional information concerning the progress or result of the function.

To activate the function specified by the High Level Function's arguments, one Application Command is directed to a certain Service in another IE. The interface to Services is defined only as functionality that is invoked by (incoming) Application Commands. How or where (e.g., by which Service) Application Commands are generated is not defined by the SPI. The Translation Table (Table 4.2-1, page 31-34) shows the relationship between High Level Functions and Application Commands. The conversion of the Application Command responses to the High Level Function responses is the responsibility of the implementation.

The definition of the Application Commands, including the protocol, is given in Chapter 5. The actions that each Application Service shall perform on an Application Command are defined in Chapter 6. The implementation of such an action can result in further messages.

4.2 TRANSLATION TABLE

The Translation Table contains one entry for each case of the High Level Functions that has a defined implementation in Chapters 5 and 6. The Translation Table is driven by the user-oriented High Level Functions. There is always one (first) command that causes the complete High Level Function to be performed. To achieve a better representation and to save the reader from having to search in Chapters 5 and 6, the complete protocol is shown in the Translation Table.

If the initiator and the receiver of a command in a protocol are identical, a corresponding IE-internal function can replace the (network-) command.

There are Application Commands that can never be the first translated Command. For example, the DESUBMIT/SPI (to IMS) invokes a FORGET/SPI (to PBS) that is never seen as a translation of a High Level Function.

4.2.1 Explanation of the Description Form

A High Level Function (and the corresponding procedure) is always called by its name, e.g., "SUBMIT". The Application Command is always called by its name concatenated with "/SPI", e.g., "SUBMIT/SPI" or "SEND/SPI". By using this naming convention, there is no confusion between High Level Functions, Application Commands and ACR-NEMA Commands.

The term "originator" is used for the IE that originates a HLF, while the term "initiator" is used for the IE that initiates a command.

Notes on High Level Functions

S	<—>	Source-IE
S=ORIGIN	<—>	Source-IE is originating IE
S#ORIGIN	<—>	Source-IE is not originating IE
S=?	<—>	Source is to be selected by IMS
D	<—>	Destination-IE
TEMP, PERM, PRINT, EXCH	<—>	Values of the CLASS argument
SUBM, NOTKNOWN	<—>	Values of the IMS-FLAG argument

Notes on Application Commands

Variable "A/xS" has "A" as the node address and "xS" as "Service" designator. Contents of "A/xS" might be "NODE100/IMS".

I	<—>	Initiator's logical address
R	<—>	Receiver's logical address
R/xS	<—>	"xS"-Service is requested at receiver
A, B	<—>	Values for logical addresses
A/xS	<—>	"xS"-Service at logical address A
MD	<—>	MOVE destination's logical address
MD/xS	<—>	"xS"-Service is requested at the MOVE destination

S,D <—> Values for source's logical address or MOVE destination's logical address; values taken from the HLF

C <—> Value for logical address chosen by IMS for a COPY with source selection

CLASS <—> "CLASS" argument value taken from the HLF

SESSION-ID <—> "Session ID" argument value

xS <—> "xS"-Service designator

- The initiator and receiver of a command response are never mentioned.

- All arguments of the HLF are transparently passed to the receiver in the Application Command, but only those arguments that govern the translation are mentioned in the table.

Remarks for COPY

The translation of COPY has different cases that are defined by the SOURCE argument on one side and by the DESTINATION, CLASS and IMS-FLAG arguments on the other side.

A characteristic of interest for the SOURCE argument is whether the LOCATION is the same as the originator, different from the originator, or unknown to the originator. The Service for the SOURCE Data Object (or its CLASS) is not relevant.

The combination of CLASS and IMS-FLAG gives the translation to the Service for the Data Object at the DESTINATION's location:

TEMP, SUBM	<—>	PBS
PERM, SUBM	<—>	PBS
TEMP, NOTKNOWN	<—>	PRS
PERM, NOTKNOWN	<—>	PRS
PRINT, NOTKNOWN	<—>	EXS
EXCH, NOTKNOWN	<—>	EXS

If the SOURCE is a different IE from the originating IE, the above arguments are passed transparently to the Data Object's DESTINATION.

There is no difference in the translation of COPY between ACR-NEMA type GET and MOVE Data Object transfers.

- The ACR-NEMA GET type transfer is done with the value D=ORIGIN.
- The ACR-NEMA MOVE type transfer is done with the value D#ORIGIN or D=ORIGIN.

Table 4.2-1 Translation Table (Part 1)

HIGH LEVEL FUNCTION	APPLICATION COMMAND
SUBMIT	<-> SUBMIT/SPI-REQUEST (I=ORIGIN, R=A/IMS)
	SUBMIT/SPI-RESPONSE
CHANGESTATE	<-> CHANGESTATE/SPI-REQUEST (I=ORIGIN, R=A/IMS)
	CHANGESTATE/SPI-RESPONSE
FORGET	<-> DESUBMIT/SPI-REQUEST (I=ORIGIN, R=A/IMS)
for each Data Object to be forgotten	- FORGET/SPI-REQUEST (I=A, R=B/PBS) - FORGET/SPI-RESPONSE - DESUBMIT/SPI-RESPONSE
COPY(S=ORIGIN, D, TEMP, SUBM) COPY(S=ORIGIN, D, PERM, SUBM)	<-> READY/SPI-REQUEST (I=ORIGIN, R=D/PBS, CLASS, SUBM[, SESSION-ID]) READY/SPI-RESPONSE SEND/SPI REQUEST (I=ORIGIN, R=D/PBS, CLASS, SUBM[, SESSION-ID]) SUBMIT/SPI-REQUEST (I=D, R=A/IMS) SUBMIT/SPI-RESPONSE SEND/SPI RESPONSE
COPY(S#ORIGIN, D, TEMP, SUBM) COPY(S#ORIGIN, D, PERM, SUBM)	<-> MOVE/SPI-REQUEST (I=ORIGIN, R=S, MD=D/PBS, CLASS, SUBM[, SESSION-ID]) MOVE/SPI RESPONSE/PENDING READY/SPI-REQUEST (I=S, R=D/PBS, CLASS, SUBM[, SESSION-ID]) READY/SPI-RESPONSE SEND/SPI REQUEST (I=S, R=D/PBS, CLASS, SUBM[, SESSION-ID]) SUBMIT/SPI-REQUEST (I=D, R=A/IMS) SUBMIT/SPI-RESPONSE SEND/SPI RESPONSE MOVE/SPI RESPONSE/FINAL

Table 4.2-1 Translation Table (Part 2)

HIGH LEVEL FUNCTION	APPLICATION COMMAND
COPY (S=?,D,TEMP,SUBM) COPY (S=?,D,PERM,SUBM)	<-> COPY/SPI-REQUEST (I=ORIGIN,R=A/IMS, MD=D/PBS,CLASS,SUBM[,SESSION-ID])
	COPY/SPI-RESPONSE/PENDING MOVE/SPI-REQUEST (I=A,R=C,MD=D/PBS, CLASS,SUBM[,SESSION-ID]) MOVE/SPI RESPONSE/PENDING READY/SPI-REQUEST (I=C,R=D/PBS, CLASS,SUBM[,SESSION-ID]) READY/SPI-RESPONSE SEND/SPI REQUEST (I=C,R=D/PBS, CLASS,SUBM[,SESSION-ID]) SUBMIT/SPI-REQUEST (I=D,R=A/IMS) SUBMIT/SPI-RESPONSE SEND/SPI RESPONSE MOVE/SPI RESPONSE/FINAL COPY/SPI RESPONSE/FINAL
COPY (S=ORIGIN,D,TEMP, NOTKNOWN)	<-> READY/SPI-REQUEST (I=ORIGIN,R=D/PRS, CLASS,NOTKNOWN[,SESSION-ID])
COPY (S=ORIGIN,D,PERM, NOTKNOWN)	READY/SPI-RESPONSE SEND/SPI REQUEST (I=ORIGIN,R=D/PRS, CLASS,NOTKNOWN[,SESSION-ID]) SEND/SPI RESPONSE
COPY (S#ORIGIN,D,TEMP, NOTKNOWN)	<-> MOVE/SPI-REQUEST (I=ORIGIN,R=S,MD=D/PRS, CLASS,NOTKNOWN[,SESSION-ID])
COPY (S#ORIGIN,D,PERM, NOTKNOWN)	MOVE/SPI RESPONSE/PENDING READY/SPI-REQUEST (I=S,R=D/PRS, CLASS,NOTKNOWN[,SESSION-ID]) READY/SPI-RESPONSE SEND/SPI REQUEST (I=S,R=D/PRS, CLASS,NOTKNOWN[,SESSION-ID]) SEND/SPI RESPONSE MOVE/SPI RESPONSE/FINAL

Table 4.2-1 Translation Table (Part 3)

HIGH LEVEL FUNCTION	APPLICATION COMMAND
COPY (S=?,D,TEMP, NOTKNOWN)	<-> COPY/SPI-REQUEST (I=ORIGIN,R=A/IMS, MD=D/PRS,CLASS,NOTKNOWN[,SESSION-ID])
COPY (S=?,D,PERM, NOTKNOWN)	COPY/SPI-RESPONSE/PENDING MOVE/SPI-REQUEST (I=A,R=C,MD=D/PRS, CLASS,NOTKNOWN[,SESSION-ID]) MOVE/SPI RESPONSE/PENDING READY/SPI-REQUEST (I=C,R=D/PRS, CLASS,NOTKNOWN[,SESSION-ID]) READY/SPI-RESPONSE SEND/SPI REQUEST (I=C,R=D/PRS, CLASS,NOTKNOWN[,SESSION-ID]) SEND/SPI RESPONSE MOVE/SPI RESPONSE/FINAL COPY/SPI RESPONSE/FINAL
COPY (S=ORIGIN,D,PRINT, NOTKNOWN)	<-> READY/SPI-REQUEST (I=ORIGIN,R=D/EXS, CLASS,NOTKNOWN,SESSION-ID)
COPY (S=ORIGIN,D,EXCH, NOTKNOWN)	READY/SPI-RESPONSE SEND/SPI REQUEST (I=ORIGIN,R=D/EXS, CLASS,NOTKNOWN,SESSION-ID) SEND/SPI RESPONSE
COPY (S#ORIGIN,D,PRINT, NOTKNOWN)	<-> MOVE/SPI-REQUEST (I=ORIGIN,R=S,MD=D/EXS, CLASS,NOTKNOWN,SESSION-ID)
COPY (S#ORIGIN,D,EXCH, NOTKNOWN)	MOVE/SPI RESPONSE/PENDING READY/SPI-REQUEST (I=S,R=D/EXS, CLASS,NOTKNOWN,SESSION-ID) READY/SPI-RESPONSE SEND/SPI REQUEST (I=S,R=D/EXS, CLASS,NOTKNOWN,SESSION-ID) SEND/SPI RESPONSE MOVE/SPI RESPONSE/FINAL

Table 4.2-1 Translation Table (Part 4, concluded)

HIGH LEVEL FUNCTION	APPLICATION COMMAND
COPY (S=?,D,PRINT, NOTKNOWN)	<-> COPY/SPI-REQUEST(I=ORIGIN,R=A/IMS, MD=D/EXS,CLASS,NOTKNOWN,SESSION-ID)
COPY (S=?,D,EXCH, NOTKNOWN)	COPY/SPI-RESPONSE/PENDING MOVE/SPI-REQUEST (I=A,R=C,MD=D/EXS, CLASS,NOTKNOWN,SESSION-ID) MOVE/SPI RESPONSE/PENDING READY/SPI-REQUEST (I=C,R=D/EXS, CLASS,NOTKNOWN,SESSION-ID) READY/SPI-RESPONSE SEND/SPI REQUEST (I=C,R=D/EXS, CLASS,NOTKNOWN,SESSION-ID) SEND/SPI RESPONSE MOVE/SPI RESPONSE/FINAL COPY/SPI RESPONSE/FINAL
QUERY (TARGET)	<-> QUERY/SPI-REQUEST (I=ORIGIN, R=TARGET[/xS]) QUERY/SPI-RESPONSE/PENDING QUERY/SPI-RESPONSE/NUMBER OF MATCHES / "number of matches" QUERY/SPI-RESPONSE < or "MAX-RESPONSES" responses QUERY/SPI-RESPONSE/FINAL
CONFIGURE (TARGET)	<-> CONFIGURE/SPI-REQUEST(I=ORIGIN, R=1st TARGET/CFS) CONFIGURE/SPI-RESPONSE repeated for each TARGET of a target list CONFIGURE/SPI-REQUEST (I=ORIGIN, R=following TARGET/CFS) CONFIGURE/SPI-RESPONSE

4.3 TRANSLATION OF GROUPCOPY

GROUPCOPY is translated by the originating IE into a sequence of High Level COPY Functions that are enclosed by two Application Commands (GCBEGIN/SPI and GCEND/SPI). See Table 4.3-1 below.

This sequence of High Level COPY Functions defines a session. The DESTINATION IE of the GROUPCOPY Function keeps track of this session by a Session ID that the DESTINATION IE assigns and includes in the GCBEGIN/SPI-RESPONSE message.

The originating IE translates the CLASS/IMS-FLAG combination to the appropriate Service (xS). If the Service is EXS, one of the transmitted Data Objects could contain a format description.

Table 4.3-1 GROUPCOPY Translation Table

GROUPCOPY (DESTINATION, CLASS, IMS-FLAG, SOURCE-1, OBJ-1-UID, ... ,SOURCE-n, OBJ-n-UID)	
	<->GCBEGIN/SPI(I=ORIGIN,R=DESTINATION/xS, CLASS,IMS-FLAG,n,TOTAL-SIZE,SESSION-ID)*
COPY (S=SOURCE-1, DESTINATION, CLASS,IMS-FLAG)	<-> see 4.2 for translation of COPY
...	
COPY (S=SOURCE-n, DESTINATION, CLASS,IMS-FLAG)	<->
	<-> GCEND/SPI(I=ORIGIN,R=DESTINATION/xS, CLASS,IMS-FLAG,SESSION-ID)

* The value for <TOTAL-SIZE> is to be stored in the "DATA-OBJECT-SIZE" element defined in Document 4.

CHAPTER 5

APPLICATION COMMANDS

5.1 GENERAL ASPECTS

All SPI communication consists of Application Commands. Application Commands are based on ACR-NEMA commands. For basic ACR-NEMA commands, refer to Section 5.3. Application Commands can be distinguished from basic ACR-NEMA commands by the SPI Recognition Code element in the command shadow group (group 1).

This chapter defines the set of Application Commands:

READY/SPI	SUBMIT/SPI	CHANGESTATE/SPI
SEND/SPI	DESUBMIT/SPI	GCBEGIN/SPI
MOVE/SPI	FORGET/SPI	GCEND/SPI
COPY/SPI	QUERY/SPI	CONFIGURE/SPI

These are named with "/SPI" to avoid confusion with High Level Functions or basic ACR-NEMA commands.

The action invoked at the receiver side by an incoming command is specified by the SERVICE argument of the command. Because a Service can be seen logically as a sub-address in an IE that is responsible for a special class of actions, the SERVICE argument is written as a qualifier to the receiver's logical address. For example:

SEND/SPI (A/PBS,TEMP)

stores a Data Object as an IMS-known Data Object on temporary storage.

SEND/SPI (A/EXS,PRINT)

creates a hard copy of a Data Object.

Remark:

PBS and EXS are on the same IE in this example. The action that takes place on the incoming SEND/SPI command is determined by the arguments of the command.

If more than two IEs are involved in the execution of an Application Command, Data Elements of the request message and the return status in the response message are passed transparently. For example, in a MOVE/SPI-REQUEST to a certain Service, the IE that holds the Data Object passes all additional data, including the finally addressed Service, with the SEND/SPI-REQUEST. The error status from the SEND/SPI-RESPONSE is passed transparently to the IE that originated the MOVE/SPI.

SPI defines only the interface between IEs, not the internal structure of an IE. There is no requirement about the IE-internal representation of a Data Object or of any other SPI-defined data, but SPI defines rules that indicate what the initiator of a command can expect from other IEs.

Each SPI-conforming IE shall be able to receive all commands and respond to them with at least a "REFUSED" Severity Status Code. If an IE could perform a function, but does not want to or is currently unable to do so, it shall at least return a response message.

An IE informs other IEs about its capabilities using the CONFIGURE/SPI command. The "COARSE" description of these capabilities shows which functions are available on that IE. This information consists of Service, CLASS and Application Command capabilities.

An IE with IMS, PBS or EXS Service usually can be expected to perform incoming requests for those Services. When the IE does not accept a request, it shall refuse with a status response indicating the reason.

5.2.2 SEND/SPI

SEND/SPI (INITIATOR, RECEIVER/SERVICE, UID, CLASS, IMS-FLAG
[, SESSION-ID])

A. Effect:

Transmit a copy of a Data Object to another location.

B. Protocol:

SENDER	RECEIVER	COMMAND
[A	B	READY/SPI-REQUEST]
[B	A	READY/SPI-RESPONSE]
A	B	SEND/SPI-REQUEST
B	A	SEND/SPI-RESPONSE

C. Actions:

The Data Object is sent from the <INITIATOR> to the <RECEIVER>. There is an additional copy of the Data Object after the operation.

D. Remarks:

1. The function that is performed with the Data Object is specified by the <SERVICE> qualifier (also by the <CLASS> and <IMS-FLAG> argument). Refer to Chapter 6 for this.
2. The <SESSION-ID> shall be specified if this SEND/SPI is part of a GROUPCOPY session.
3. The SEND/SPI can optionally be preceded by the READY/SPI handshake for "large" Data Objects.

E. Errors/Messages:

1. The SEND/SPI-RESPONSE status values are defined by ACR-NEMA.
2. If the SEND/SPI-RESPONSE has the Severity Status Code "REFUSED", SPI defines the following reasons for refusal that are passed with an additional Error Status Code (see also READY/SPI):
 - a. The specified Service is not implemented (SERV-NOT-IMPLEM)
 - b. The specified Service is not available (SERV-NOT-AVAIL)
 - c. The specified Service/CLASS combination is not implemented (S/C-NOT-IMPLEM)
 - d. The specified Service/CLASS combination is not available (S/C-NOT-AVAIL)
 - e. The specified Service/CLASS/IMS-FLAG combination is not implemented (S/C/F-NOT-IMPLEM)
 - f. The specified Service/CLASS/IMS-FLAG combination is not available (S/C/F-NOT-AVAIL)
 - g. The specified size cannot be accepted now (SIZE-TOO-BIG)
 - h. The Data Object announced in a READY/SPI-REQUEST already exists at the receiver (UID-EXIST)
3. The SEND/SPI-RESPONSE has the Error Status Code "INVALID-SESS-ID" if there is no opened GROUPCOPY session for the specified <SESSION-ID>.
4. The SEND/SPI-RESPONSE has the Error Status Code "SESS-TIMEOUT" if a timeout occurred at <RECEIVER> during a GROUPCOPY session.
5. The SEND/SPI-RESPONSE has the Error Status Code "SESS-ID-MISSING" if there is no specified <SESSION-ID> in the request and a <SESSION-ID> is expected by the <INITIATOR>.
6. The SEND/SPI-RESPONSE has the additional Error Status Code "SUBMIT-FAILED" if PBS was not able to submit the Data Object. PBS shall delete the Data Object in this case.
7. For the first SEND/SPI-REQUEST in a GROUPCOPY session the following errors might occur:
 - a. The specified medium is not available (MEDIUM-NOT-AVAIL)
 - b. The specified Format is not available (FORMAT-NOT-AVAIL)
 - c. There is no Format specified (NO-FORMAT-SPEC)

5.2.3 MOVE/SPI

MOVE/SPI (INITIATOR, RECEIVER, MOVE-DESTINATION/SERVICE, UID, CLASS,
IMS-FLAG [,SESSION-ID])

A. Effect:

Transfer a Data Object from the RECEIVER to another location.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B	MOVE/SPI-REQUEST (DESTINATION=C/XS)
B	A	MOVE/SPI-RESPONSE/PENDING
[B	C	READY/SPI-REQUEST]
[C	B	READY/SPI-RESPONSE]
B	C	SEND/SPI-REQUEST
C	B	SEND/SPI-RESPONSE
B	A	MOVE/SPI-RESPONSE/FINAL

C. Actions:

1. The <INITIATOR> asks the <RECEIVER> to send the Data Object identified by <UID> to the <SERVICE> in the <MOVE-DESTINATION>.
2. If the Data Object is available at <RECEIVER>, <RECEIVER> sends it to <MOVE-DESTINATION>'s <SERVICE>. Otherwise an error situation occurs.
3. If the MOVE/SPI is part of a GROUPCOPY session, the optional Data Element <SESSION-ID> is specified. <SESSION-ID> will be passed transparently to the <MOVE-DESTINATION/SERVICE>.

D. Remarks:

1. <MOVE-DESTINATION> may be identical to <INITIATOR>.
2. Because the storage-class from which the <RECEIVER> retrieves the Data Object is not relevant here, there is no Service qualifier for the <RECEIVER>.
3. The <MOVE-DESTINATION/SERVICE> specifies the logical address and the required Service at this address.

The <SERVICE>, <CLASS> and <IMS-FLAG> arguments specify what the DESTINATION IE shall do with the Data Object.

4. Responses that occur during the SEND/SPI sequence shall be returned to the originator transparently with the MOVE-RESPONSE/FINAL status. (The "FINAL" status is any status other than "PENDING".)

E. Errors/Messages:

1. See the ACR-NEMA Standard.
2. If <RECEIVER> does not want to transmit the Data Object, <RECEIVER> adds the Error Status Code "TRANSMISS-REJ" in the MOVE/SPI-RESPONSE/FINAL.

5.2.4 SUBMIT/SPI

SUBMIT/SPI (INITIATOR, RECEIVER/IMS, UID, DESC [, OPTIONAL-ELEMENTS]
[, DESC-SPECIFICATION])

A. Effect:

Make an SPI-conforming Data Object known to IMS.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B/IMS	SUBMIT/SPI-REQUEST
B	A	SUBMIT/SPI-RESPONSE

C. Actions:

1. IMS makes its entries for the submitted copy of the Data Object in its list upon the SUBMIT/SPI-REQUEST.
2. The submitting IE sets the "IMS-known" attribute of the submitted Data Object to "YES" upon the SUBMIT/SPI-RESPONSE.
3. See the High Level Function (3.3.1) for further details.

D. Remarks:

1. A copy of the Data Object shall be resident at the IE that issues the SUBMIT/SPI.
2. An IE shall not change or delete a Data Object while it is IMS-known.
3. The receiving Service of a SUBMIT/SPI-REQUEST is always IMS.
4. The <INITIATOR> of a SUBMIT/SPI-REQUEST shall have PBS.
5. The optional <DESC-SPECIFICATION> argument shall be used to specify the DESC elements of Data Objects of a type that is not defined by Document 4.

E. Errors/Messages:

1. The Data Object already is IMS-known at this location for the IMS. IMS returns the "PREV-SUBMITTED" Error Status Code.
2. When the <UID> is unspecified, IMS refuses the SUBMIT/SPI and returns the Error Status Code "UID-MISSING".
3. When IMS currently is unable to perform the SUBMIT/SPI, IMS returns the Severity Status Code "REFUSED".

5.2.5 CHANGESTATE/SPI

CHANGESTATE/SPI (INITIATOR, RECEIVER/IMS, UID, STATE)

A. Effect:

Change the State of an IMS-known Data Object.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B/IMS	CHANGESTATE/SPI-REQUEST
B	A	CHANGESTATE/SPI-RESPONSE

C. Actions:

IMS changes the State of the Data Object.

D. Remarks:

IMS may refuse to perform this function depending on which <STATE> is to be changed and which IE issued this request.

E. Errors/Messages:

1. If the <UID> is unspecified, IMS refuses the CHANGESTATE/SPI and returns the Error Status Code "UID-MISSING".
2. If IMS does not know <UID>, the Severity Status Code "INFO-NOT-AVAIL" is returned.
3. If the CHANGESTATE/SPI would result in an overflow for non-SPI-defined State_labels, the Error Status Code "STATE-OVERFLOW" is returned.
4. The <INITIATOR> is not allowed to change this <STATE>, because it would violate SPI-defined or private rules. In this case the Error Status Code "STATE-VIOLATION" is returned.

5.2.6 DESUBMIT/SPI

DESUBMIT/SPI (INITIATOR, RECEIVER/IMS, UID, LOCATION, FORGET-LAST-COPY)

A. Effect:

Remove one or more copies of a Data Object from IMS knowledge.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B/IMS	DESUBMIT/SPI-REQUEST
B	C/PBS	FORGET/SPI-REQUEST
C	B	FORGET/SPI-RESPONSE
B	D/PBS	FORGET/SPI-REQUEST
D	B	FORGET/SPI-RESPONSE
:	:	Repetition for each Data Object copy to be forgotten
B	A	DESUBMIT/SPI-RESPONSE

C. Actions:

1. The <INITIATOR> specifies with <LOCATION>, at which location(s) the Data Object <UID> should be forgotten. <LOCATION> can be set to "ALL". The <FORGET-LAST-COPY> flag can be set to "NO" if the originator wants at least one copy of the Data Object to be IMS-known.
2. If the <LOCATION> list contains all locations where IMS-known copies are stored, but the <FORGET-LAST-COPY> flag is "NO", IMS deletes an arbitrary location from the list. If the resulting list is empty, an error response occurs.
3. For each entry in the <LOCATION> list, IMS sends a FORGET/SPI REQUEST (PBS) to those IEs that store the Data Object.
4. For each positive FORGET/SPI-RESPONSE, IMS erases its knowledge about that copy.

D. Remarks:

1. If an IE with PBS has limited storage capacity and wants to delete its copy of an IMS-known Data Object, it should set the <FORGET-LAST-COPY> flag to "NO". IMS will only allow the DESUBMIT/SPI if there is another IMS-known copy of the Data Object.
2. IMS is not required to issue FORGET/SPI-REQUESTS for a copy of a Data Object on permanent storage.

E. Errors/Messages:

1. If <RECEIVER> does not know <UID>, it returns the Severity Status Code "INFO-NOT-AVAIL".
2. <RECEIVER> returns the location(s) where each Data Object copy is forgotten, and returns the reason for each location where the Data Object copy is not forgotten:
 - a. The Data Object is on permanent storage and is not forgotten/deleted (PERM-STORAGE)
 - b. The location is currently not available (LOC-NOT-AVAIL)
 - c. The copy of the Data Object is the last copy (LAST-COPY)

5.2.7 FORGET/SPI

FORGET/SPI (INITIATOR, RECEIVER/PBS, UID)

A. Effect:

IMS causes an IE to make its copy of a Data Object IMS-unknown.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B/PBS	FORGET/SPI-REQUEST
B	A	FORGET/SPI-RESPONSE

C. Actions:

1. If PBS accepts the request to forget the Data Object, PBS sets the status of the Data Object to IMS-unknown.
2. IMS takes its Data Object entries for <RECEIVER's PBS> from its lists upon a positive FORGET/SPI-RESPONSE (PBS).

D. Remarks:

1. Only IMS may issue an FORGET/SPI-REQUEST.
2. Only PBS can be expected to accept a FORGET/SPI-REQUEST.
3. PBS may refuse the FORGET/SPI. If this occurs, the Data Object remains IMS-known.

E. Errors/Messages:

1. If <RECEIVER> does not know <UID>, it returns the Severity Status Code "INFO-NOT-AVAIL".
2. <RECEIVER> returns the Severity Status Code "SUCCESS" if the FORGET/SPI was performed.
3. If the Data Object is on permanent storage and is not forgotten, <RECEIVER> returns "PERM-STORAGE".

5.2.8 QUERY/SPI

QUERY/SPI (INITIATOR, RECEIVER[/SERVICE], TEMPLATE, LOCATION, MAX-RESPONSES, RETURN-INFO-FLAG, COUNT)

A. Effect:

Obtain information about Data Objects and other Data Elements known by the target IE.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B	QUERY/SPI-REQUEST
[B	A	QUERY/SPI-RESPONSE/PENDING]
		0...n times
B	A	QUERY/SPI-RESPONSE/FINAL

C. Actions:

1. The <RECEIVER> may return a first QUERY/SPI-RESPONSE/PENDING without a match to inform the <INITIATOR> that the QUERY/SPI is accepted.
2. The <RECEIVER> returns information for each Data Object that matches <TEMPLATE> (and <LOCATION>). Each match is returned in one QUERY/SPI-RESPONSE.
3. If there are no more matches, or <MAX-RESPONSES> are met, the <RECEIVER> returns the QUERY/SPI-RESPONSE/FINAL. (The "FINAL" status is any status other than "PENDING".)

D. Remarks:

1. Details are described in the High Level Functions (3.3.5)
2. If <RECEIVER/SERVICE> is specified, the QUERY/SPI is directed to the requested <SERVICE> in the <RECEIVER>. This returns all information known to the <SERVICE>.

If only <RECEIVER> is specified (i.e., SERVICE is omitted), <RECEIVER> shall return information about all information known to each Service in the <RECEIVER>. For example, an IE that has PRS and PBS can return information about both storage-classes upon a QUERY/SPI if no special SERVICE is specified. Such implementations are optional in the SPI.

The only Service required by the SPI to accept a QUERY/SPI is IMS. IMS returns knowledge about IMS-known Data Objects in each PBS of those IEs specified by <LOCATION>; it may optionally be able to provide other information as well.

E. Errors/Messages:

1. If the <TEMPLATE> does not satisfy the SPI-defined syntax, <RECEIVER> returns a QUERY/SPI-RESPONSE/FINAL with the additional Error Status Code "ILLEGAL-TEMPLATE".
2. If the <TEMPLATE> includes operands/Data Elements that are not supported by the <RECEIVER>, <RECEIVER> returns a QUERY/SPI-RESPONSE/FINAL with the additional Error Status Code "TEMPLATE-NOT-SUPP".
3. If <RECEIVER> already returned <MAX-RESPONSES> matches in QUERY/SPI-RESPONSES, the QUERY/SPI-RESPONSE/FINAL has the additional Error Status Code "MATCH-OVERFLOW".

5.2.9 COPY/SPI

COPY/SPI (INITIATOR, RECEIVER/IMS, UID, DESTINATION/SERVICE, CLASS, IMS-FLAG [, SESSION-ID])

A. Effect:

An IE asks IMS to get a copy of an IMS-known Data Object moved from any location's PBS.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B	COPY/SPI-REQUEST (IMS)
B	A	COPY/SPI-RESPONSE/PENDING
B	C	MOVE/SPI-REQUEST (DESTINATION=D/xS)
C	B	MOVE/SPI-RESPONSE/PENDING
[C	D	READY/SPI-REQUEST (xS)]
[D	C	READY/SPI-RESPONSE]
C	D	SEND/SPI-REQUEST (xS)
D	C	SEND/SPI-RESPONSE
C	B	MOVE/SPI-RESPONSE/FINAL
B	A	COPY/SPI-RESPONSE/FINAL

C. Actions:

1. IMS searches its lists for the Data Object identified by <UID>.
2. IMS issues a MOVE/SPI-REQUEST (xS) to an IE with PBS that stores the Data Object. IMS passes the <DESTINATION/SERVICE> argument of the COPY/SPI REQUEST transparently.
3. The IE selected by IMS in Action 2 (C) sends a copy of the Data Object to <DESTINATION/SERVICE>.

D. Remarks:

1. The receiver of a COPY/SPI-REQUEST is always IMS.
2. The <DESTINATION> may be the same IE as the <INITIATOR>.

E. Errors/Messages:

1. If IMS does not know <UID>, it returns the Severity Status Code "INFO-NOT-AVAIL".
2. For other errors, see MOVE/SPI, SEND/SPI, and READY/SPI.

5.2.10 CONFIGURE/SPI

CONFIGURE/SPI (INITIATOR, RECEIVER/CFS, MY-CAPABILITIES, RESPONSES-FLAG
[, CAPABILITY])

A. Effect:

Exchange capability information with another IE.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B/CFS	CONFIGURE/SPI-REQUEST
B	A	CONFIGURE/SPI-RESPONSE

C. Actions:

1. The <INITIATOR> informs the <RECEIVER> about the <INITIATOR>'s capabilities <MY-CAPABILITIES> that are currently available.
2. The <RECEIVER> informs the <INITIATOR> about the <RECEIVER>'s capabilities corresponding to the <RESPONSES-FLAG> specification that is set to "NONE", "COARSE" or "FINE".

D. Remarks:

1. The "COARSE" information, which is also transmitted by <MY-CAPABILITIES> in the same format, gives information about the IE's:
 - a. Services (IMS, PBS, PRS, EXS)
 - b. Classes (TEMP, PERM, PRINT and EXCH)
 - c. Application Commands that are accepted by the IE
 - d. Application Commands that are issued by the IE
 - e. ACR-NEMA Commands that are accepted and/or issued.
2. This command is only applicable to the Configuration Service, and it is the only command of the Configuration Service.
3. The <INITIATOR> and <RECEIVER> are responsible for what each will do with the capability information exchanged.
4. If an IE goes off-line in a controlled way, it shall transmit an empty capability-list <MY-CAPABILITIES> before exiting.

E. Errors/Messages:

An IE that cannot support "FINE" CONFIGURE/SPI-REQUESTs for the requested <CAPABILITY> returns the Severity Status Code "REFUSED" with an Error Status Code:

- a. The requested capability is not implemented (CAP-NOT-IMPLEM)
- b. No Fine Capability information for the requested capability is available (NO-FINE-INFO)

5.2.11 GCBEGIN/SPI

GCBEGIN/SPI (INITIATOR, RECEIVER/SERVICE, CLASS, IMS-FLAG,
NUM-OBJECTS, TOTAL-SIZE, SESSION-ID)

A. Effect:

Start a GROUPCOPY session.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B	GCBEGIN/SPI-REQUEST
B	A	GCBEGIN/SPI-RESPONSE

C. Actions:

The <RECEIVER/SERVICE> starts a GROUPCOPY session. The whole GROUPCOPY session is identified by <SESSION-ID> that is assigned by the <RECEIVER>.

D. Remarks:

The value of the <TOTAL-SIZE> argument is an estimate of the expected total size.

E. Errors/Messages:

1. The GCBEGIN/SPI-RESPONSE returns the Severity Status Code "SUCCESS" and the <SESSION-ID> if it was successfully performed. This response specifies a time period for which the <RECEIVER> will guarantee the required resources for the session.
2. The GCBEGIN/SPI-RESPONSE has the Severity Status Code "REFUSED", with the additional Severity Status Code "WAIT" if the IE is currently unable to start the session, but expects to be able to start soon. In such a case, the IE shall indicate an estimate of the required delay.
3. The GCBEGIN/SPI-RESPONSE indicates the reason for an error with the following Error Status Codes:
 - a. The specified Service is not implemented (SERV-NOT-IMPLEM)
 - b. The specified Service is not available (SERV-NOT-AVAIL)
 - c. The specified Service/CLASS combination is not implemented (S/C-NOT-IMPLEM)
 - d. The specified Service/CLASS combination is not available (S/C-NOT-AVAIL)
 - e. The specified Service/CLASS/IMS-FLAG combination is not implemented (S/C/F-NOT-IMPLEM)
 - f. The specified Service/CLASS/IMS-FLAG combination is not available (S/C/F-NOT-AVAIL)
 - g. The specified NUM-OBJECTS cannot be accepted now (TOO-MANY-OBJECTS)

5.2.12 GCEND/SPI

GCEND/SPI (INITIATOR, RECEIVER/SERVICE, SESSION-ID
[, COMPLETION-STATE])

A. Effect:

Terminate a GROUPCOPY session.

B. Protocol:

SENDER	RECEIVER	COMMAND
A	B	GCEND/SPI-REQUEST
B	A	GCEND/SPI-RESPONSE

C. Actions:

The <RECEIVER> terminates the GROUPCOPY session.

D. Remarks:

<COMPLETION-STATE> can have the value "GO ON" to inform the <RECEIVER> that the session is successfully completed. It has the value "CANCEL" if <INITIATOR> wants the complete session to be cancelled. The reaction to such a cancellation is the responsibility of the <RECEIVER>.

E. Errors/Messages:

1. The GCEND/SPI-RESPONSE has the Error Status Code "SESS-ID-UNKNOWN" if the GROUPCOPY session was not started correctly.
2. The GCEND/SPI-RESPONSE has the Error Status Code "SESS-TIMEOUT" if a timeout occurred at <RECEIVER>.

5.3 HANDLING OF ACR-NEMA COMMANDS

The network communication in an SPI network can be separated into two domains: the SPI domain and the outside-SPI domain.

Inside the SPI domain, Application Commands are used between IEs. Since there is an SPI Recognition Code in all Application Commands, these commands can be distinguished from outside-SPI commands.

Inside the SPI domain, it is allowable to use basic ACR-NEMA commands only in those cases where there is no corresponding Application Command. Therefore, ECHO, CANCEL, DIALOG and FIND may be used inside the SPI domain.

ACR-NEMA commands received from IEs outside the SPI domain shall be properly processed by IEs inside the SPI domain.

ACR-NEMA data sets created outside the SPI domain are "private" when they enter the SPI Domain. SPI does not specify what an IE that receives such a data set will do with it. For example, an archive IE may make it into an IMS-known Data Object, while a viewing station might not.

Those SPI IEs that transfer Data Objects to, and ACR-NEMA data sets from, the outside-SPI domain are responsible for adding and deleting the SPI-defined UID and other SPI Data Elements. Non-SPI IEs cannot be expected to provide the SPI UID; therefore, when an SPI IE receives a data set from a non-SPI IE, the SPI IE is responsible for adding a UID and other necessary SPI Data Elements required to make the data set into an SPI-conforming Data Object at that time. If the addition of SPI Data Elements would cause collisions with already existing elements, the result is unspecified by SPI. Similarly, when an SPI IE transmits a copy of a Data Object to a non-SPI IE, the SPI IE is responsible for removing the UID and other SPI Data Elements from the transmitted copy.

Data Objects created inside the SPI domain can be distinguished from data sets created outside the SPI domain by the SPI Recognition Code in the SPI-created Data Objects. (Strictly speaking, there is no guarantee that an IE outside the SPI domain cannot include such an element in a data set.)

5.4 MAPPING APPLICATION COMMANDS TO ACR-NEMA

The Application Commands are ACR-NEMA commands with additional Data Elements in Group 1, the shadow group of ACR-NEMA command group 0. Refer to SPI Document 4 for exact definitions.

5.5 APPLICATION COMMANDS / APPLICATION SERVICES

The set of Application Commands and their applicability to Application Services are shown in Table 5.5-1 below.

This table shows which commands can be sent to an Application Service as the receiver of the command, and thus, also defines the functions an Application Service offers to the participants in the SPI network.

Because Document 3 does not define how and where the Application Commands are generated, this table does not need an extension that shows which Service issues what Command.

Table 5.5-1 Application Commands Accepted by Application Services

APPLICATION COMMAND	S E R V I C E				
	IMS	PBS	PRS	EXS	CFS
COPY/SPI	M	-	-	-	-
SUBMIT/SPI	M	-	-	-	-
CHANGESTATE/SPI	M	-	-	-	-
DESUBMIT/SPI	M	-	-	-	-
QUERY/SPI	M	O	O	-	-
READY/SPI	-	M	O	M	-
SEND/SPI	-	M	O	M	-
MOVE/SPI	-	M	O	-	-
FORGET/SPI	-	M	-	-	-
GCBEGIN/SPI	-	O	O	O	-
GCEND/SPI	-	O	O	O	-
CONFIGURE/SPI	-	-	-	-	M

Interpretation: M(andatory): Service shall accept the command.
 O(ptional) : Service is allowed, but not required to accept the command.
 - : Undefined by the SPI.

CHAPTER 6

APPLICATION SERVICES

6.1 INTRODUCTION

Application Services are a set of capabilities that are optionally provided by an IE. These capabilities can be separated into Data Object storage capabilities, Data Object management capabilities and network/system configuration capabilities. An Application Service implements a class of related functions, e.g., storing, retrieving and deleting Data Objects. Therefore, a Service is also a conceptual part of an IE that can be addressed by an IE's logical name and the Service.

Each Application Service is defined by a set of Application Commands. SPI does not specify the implementation or the internal software/hardware-architecture for any Service. An IE in an SPI network can use the functionality of an Application Service in another IE by issuing the Application Commands of this Service.

6.2 IMAGE MANAGEMENT SERVICE

One added functionality of SPI compared to basic ACR-NEMA is the Image Management Service (IMS). IMS provides a unique directory of Data Objects in one SPI Network. Therefore, IMS is resident on at most one IE in one SPI network.

IMS manages IMS-known Data Objects in the sense of a directory. The storage of these IMS-known Data Objects is independent from the management function and is handled in IEs with Public Storage Service (PBS). IMS also keeps track of the status and location of the Data Object.

A Data Object becomes IMS-known when it is announced to the IMS. A Data Object may be IMS-known for more than one copy in more than one location (IE). SPI assures a defined behavior for IMS-known Data Objects to the user whereby IMS-known Data Objects can be seen as "public". Copies of IMS-known Data Objects can be "forgotten", making them IMS-unknown again.

IMS shall accept these commands and take corresponding actions:

Command	Action(s)
SUBMIT/SPI	A (copy of a) Data Object becomes IMS-known.
CHANGESTATE/SPI	IMS changes the recorded State value for an IMS-known Data Object.
DESUBMIT/SPI	One or more copies of a Data Object become IMS-unknown.
QUERY/SPI	IMS returns information about IMS-known Data Objects.
COPY/SPI	IMS searches its lists for the location of an IMS-known Data Object and initiates the transmission of a copy of the Data Object.

An IE with IMS shall be able to issue these commands:

FORGET/SPI
MOVE/SPI

6.3 PUBLIC STORAGE SERVICE

Public Storage Service (PBS) is the capability of storing IMS-known Data Objects. PBS provides temporary and/or permanent storage. Data Objects stored on PBS are accessible to other IEs. A Data Object stored on PBS shall not be deleted while it is IMS-known.

PBS determines whether the Data Object is to be stored on temporary or permanent storage by the value of the CLASS attribute of the SEND/SPI REQUEST to PBS.

A Public Storage Service shall issue a SUBMIT/SPI-REQUEST to IMS for each Data Object that is sent to PBS with a SEND/SPI having the IMS-FLAG set to "SUBM". The SUBMIT/SPI-REQUEST makes the Data Object IMS-known.

An IE with PBS can be expected to send IMS-known Data Objects to another IE on request.

PBS may have the following reasons to get rid of a Data Object: Public Storage Service may have a limited temporary storage capacity. A Data Object stored on a certain IE's PBS on temporary storage can only be expected to be stored at that location for a limited period of time. A Data Object stored on permanent storage is expected to be stored permanently, but those Data Objects may age because of a limited archiving period. Data Objects could also be archived erroneously.

PBS shall issue a DESUBMIT/SPI-REQUEST to the IMS to make a Data Object IMS-unknown before it can delete it. Since a Public Storage Service is supposed to work automatically, it should set the FORGET-LAST-COPY flag to "NO" if it issues a DESUBMIT/SPI-REQUEST to IMS. If IMS does not accept the DESUBMIT/SPI, PBS can transmit a copy of the Data Object to another PBS and try the DESUBMIT/SPI again.

PBS can be caused by IMS to set the "IMS-known" attribute to "NO" if IMS issues a FORGET/SPI-REQUEST to PBS. PBS is free in its decision whether or not to accept the FORGET/SPI-REQUEST. For example, a Data Object on permanent storage is not intended to be forgotten. However, PBS shall implement the FORGET/SPI function if the design of PBS requires it to issue DESUBMIT/SPI-REQUESTs to the IMS.

PBS shall accept these commands:

READY/SPI, SEND/SPI, MOVE/SPI, FORGET/SPI

PBS is allowed, but not required, to accept these commands:

QUERY/SPI, GCBEGIN/SPI, GCEND/SPI

An IE with PBS shall be able to issue these commands:

SUBMIT/SPI, READY/SPI, SEND/SPI

An IE with PBS can issue this command:

DESUBMIT/SPI

6.3.1 Remarks on Commands to PBS

SEND/SPI Store the Data Object as an IMS-known Data Object.

The Data Object is sent to PBS and stored on temporary or permanent storage according to the CLASS argument.

If the IMS-FLAG argument is set to "SUBM", PBS issues the SUBMIT/SPI to IMS. If the Data Object is stored on permanent storage, PBS transmits the "ARCHIVED" State "YES" after a successful SUBMIT/SPI with a subsequent CHANGESTATE/SPI.

If PBS initiates SUBMIT/SPIs, the following protocol applies:

SENDER	RECEIVER	COMMAND
A	B/PBS	SEND/SPI-REQUEST
B	C/IMS	SUBMIT/SPI-REQUEST
C	B	SUBMIT/SPI-RESPONSE
B	A	SEND/SPI-RESPONSE

When the SUBMIT/SPI-RESPONSE arrives, PBS sets the "IMS-known" attribute to "YES". The deletion rules for IMS-known Data Objects apply now.

If the SUBMIT/SPI fails, PBS deletes the Data Object and returns an appropriate SEND/SPI-RESPONSE status.

FORGET/SPI IMS causes PBS to make a Data Object IMS-unknown.

PBS can refuse this request; it can be expected to do so for Data Objects on permanent storage.

QUERY/SPI Obtain information about Data Objects known by PBS.

6.4 PRIVATE SERVICE

Private Service (PRS) is the capability of receiving/sending IMS-unknown Data Objects. PRS is the minimum Service in an IE to receive Data Objects from (or to send Data Objects to) other Services (PRS, PBS, EXS) in any other IE.

The copy of a Data Object sent to PRS is IMS-unknown at the receiver: it is "private". SPI does not require any particular functionality for Data Objects on PRS.

How and whether PRS provides temporary and/or permanent storage is the responsibility of the implementation. SPI does not define the behavior of "private temporary storage" and "private archive", but SPI allows PRS to serve these purposes.

A Data Object sent to another IE's PRS results in an IMS-unknown copy of the Data Object at the receiver. If the receiver wants the Data Object to become IMS-known, the receiver can submit it. In this case, the Data Object would be in PBS after the submit.

There are no mandatory commands for PRS, because PRS is "private".

PRS is allowed, but not required, to accept these commands:

READY/SPI, SEND/SPI, MOVE/SPI, QUERY/SPI, GCBEGIN/SPI, GCEND/SPI

6.4.1 Remarks on Commands to PRS

SEND/SPI	PRS receives the Data Object as an IMS-unknown Data Object.
	SPI does not require the receiver to perform any specific function on this IMS-unknown copy of the Data Object.
MOVE/SPI	This command is intended to recall the Data Object and send it to the requested Service at the MOVE-DESTINATION. PRS may refuse the processing of this command, because no IE can be forced to transmit a "private" Data Object.
QUERY/SPI	PRS returns information about the Data Objects it knows about.

6.5 EXPORT SERVICE

Export Service (EXS) is the capability of creating copies of Data Objects on off-line media. EXS provides the creation of hard copies and/or off-line data exchange volumes.

Export Service is generally driven by a High Level GROUPCOPY Function type operation. An export session starts with a starting message, can go on with the format specification of the export session, continues with the transmission of Data Objects and stops with a final message. Export Service identifies the whole session by the SESSION-ID that EXS assigns in the GCBEGIN/SPI-RESPONSE message.

There can be types of EXS implementations that allow a "one-Data-Object-session" with a default format. These can be driven without the GROUPCOPY mechanism.

The set of Data Objects on an off-line medium is intended to be exported out of the SPI-network where it was created.

Export Service is defined for two classes: "PRINT" and "EXCH" (exchange). Print CLASS is defined for the creation of hard copies; exchange CLASS for the creation of off-line data exchange volumes.

One Data Object transmitted in the export session typically contains a format specification. This can be the first transmitted Data Object. For a print job, this might specify the medium type (film, paper, slides, etc.), the arrangement order, and other attributes. For an exchange job, this might specify the medium type (floppy, optical disk, etc.) and the volume specification.

Each model of a hard-copy unit can be assumed to implement a different set of functions, formatting, etc. Therefore, SPI does not define an SPI format for printing, but only defines the way such a format can be communicated. The hard-copy format types may be obtained with the CONFIGURE Function, so an IE knows which format specifications are applicable.

EXS shall accept these commands and take the corresponding actions:

READY/SPI	Establish agreement about a data transfer.
SEND/SPI	Receive a format description.
	Receive the Data Object and put it on the off-line medium in accordance with the rules of the export format specified.

EXS can accept these commands and take the corresponding actions:

GCBEGIN/SPI	Start an export session. Assign a session-ID.
GCEND/SPI	Terminate an export session.

6.6 CONFIGURATION SERVICE

Configuration Service (CFS) manages the information exchange between IEs about the Service, storage and command capabilities of these IEs. There is a "COARSE" and a "FINE" configuration capability.

The "COARSE" configuration information returned contains basic information indicating which functions the IE offers to other network participants via Application Commands.

The "FINE" configuration information returned contains either the formats applicable for EXS or detailed information about the query capability of an IE.

CFS shall accept this command and perform the corresponding action:

CONFIGURE/SPI Inform another IE's CFS about the capabilities of the originating IE. The other IE's CFS can be requested to respond with its capabilities.

INDEX

- A -

ACR-NEMA commands 56

Application:

Command 3, 16, 28, 36*, 37,

38*, 57*, 58

Service 5, 10*, 35-37, 57-58*

Service name 10, 30

<ASCII Numeric String> 24*

<ASCII Text String> 24*

- C -

CANCEL (ACR) 56

<CAPABILITY> 27, 52-53

CFS = Configuration Service

CFS command 64*

CHANGESTATE (HLF) 13, 14*, 19*, 31

CHANGESTATE/SPI 31, 36, 45*, 57,

59, 61

<CLASS> 21-22, 26, 29-35, 37-42,

51, 54, 61, 63

Coarse Capabilities (DE) 15, 27,

52

command:

mapping (SPI) 56*

refusal 37*

request 28*

response 28*

shadow group 36

<COMPLETION-STATE> 55

Configuration Service 10, 52-53,

57, 64*

CONFIGURE (HLF) 13, 15*, 27*,

34*, 63

CONFIGURE/SPI 34, 36-37, 52*, 53,

57, 64

CONFIGURE/SPI-RESPONSE:

"COARSE" 52*, 64

"FINE" 52*, 64

COPY (HLF) 13, 14*, 15, 21*, 26,

30-31, 32-34*

COPY/SPI 32-34, 36, 51*, 57, 59

<COUNT> 23, 25, 49

- D -

Data Object 6*

classification 6*

conformance rules 6, 7*

Description 7-8

identification 8

IMS-known 5, 8, 13, 18*, 19-20,

21*, 23, 58, 60

IMS-unknown 5, 8, 13, 18*,
19-20, 21*, 23, 58, 60, 62

location 9*

modification 8*

modification history 8*

Type 5

<DATA-OBJECT-SIZE> 38

DATA-OBJECT-SIZE (DE) 35

Data Set Type 7*

DESC:

= Data Object Description

elements 23, 44

free-formatted 8

information 25

<DESC> 18, 44

<DESC-SPECIFICATION> 38, 40, 44

<DESTINATION> 16, 21, 26, 30, 35,

51

DESUBMIT/SPI 31, 36, 46*, 47, 57,

59-60

DIALOG (ACR) 56

domain:

outside-SPI 56*

SPI 56*

- E -

ECHO (ACR) 56

Element 24*

<Even Integer> 24*

Export Service 10, 57, 63*

EXS = Export Service

EXS commands 63*

- F -

FIND (ACR) 15, 56

Fine Capability (DE) 15

FORGET (HLF) 13, 14*, 18, 20*,

22, 31

<FORGET-LAST-COPY> 20*, 46, 60

FORGET/SPI 31, 36, 46*, 48*, 57,

59-61

- G -

GCBEGIN/SPI 35-36, 54*, 57, 60,

62-63

GCEND/SPI 35-36, 55*, 57, 60,

62-63

GET (ACR) 14, 30

Group 24*

GROUPCOPY (HLF) 13, 15*, 26*,
35*, 63
session 15, 38, 40-41, 54-55

- H -

hard copy:
output 10
print 26, 36*
High Level Function 11*, 13*, 16*
HLF:
= High Level Function
argument conventions 16*
description format 16*
"initiator" 29*
"originator" 29*
Translation Table 28-35*

- I -

Identifier, Unique 7, 8*, 56*
image exchange format 3
Image Management Service 5, 8,
10, 13-15, 44-46, 49, 51, 57,
58*
Data Object registration 5,
13-14*, 17-18*
Imaging Equipment 3, 5, 16
logical model 11*
non-SPI-conforming 12*
SPI-conforming 6, 12*
IMS = Image Management Service
IMS commands 59*
<IMS-FLAG> 21-22, 26, 29-30, 35,
38-42, 51, 54, 60-61
<INITIATOR> 38-42, 44-46, 48-49,
51-52, 54-55
<Integer> 24*

- L -

<LOCATION> 16, 20, 23, 30, 46-47,
49
logical address of an IE 9, 16

- M -

<MAX-RESPONSES> 23, 25*, 34, 49
message 5, 28*
MOVE (ACR) 14, 30
<MOVE-DESTINATION> 42, 62
MOVE/SPI 31-34, 36, 42*, 43, 51,
57, 59-60, 62
<MY-CAPABILITIES> 27*, 52

- N -

name, logical = logical address
<NUM-OBJECTS> 54

- O -

<OBJ-n-UID> 26, 35
<Odd Integer> 24*
off-line media 10, 63
off-line status 27*
on-line status 27*
<OPTIONAL-ELEMENTS> 18, 44
outside-SPI domain 56*

- P -

PBS = Public Storage Service
PBS commands 60-61*
permanent storage 9, 14-15, 20, 45
Private Service 10, 57, 62*
Private Storage Service 49
PRS = Private Service
PRS commands 62*
Public Storage Service 10, 48-49,
51, 57, 58*, 60*

- Q -

QUERY (HLF) 13-15, 23-25*, 34*
response "COARSE" 15, 27*
response "FINE" 15, 27*
syntax 24*
<TEMPLATE> argument 23, 24*
QUERY/SPI 34, 36, 49*, 50, 57,
59-62

- R -

READY/SPI 31-34, 36, 38*, 39-42,
51, 57, 60, 62-63
<RECEIVER> 38-52, 54-55
<RESPONSES-FLAG> 27, 52
<RETURN-INFO-FLAG> 23*, 25*, 49

- S -

SEND (ACR) 14
SEND/SPI 31-34, 36, 40-41*, 42,
51, 57, 60-63
<SERVICE> 35-36, 38-42, 49, 51,
54-55
<SESSION-ID> 31-34, 35*, 38-42,
51, 54-55, 63

<Shadow Owner Code> 24*
<SOURCE> 16, 21, 30, 51
<SOURCE-n> 26, 35

SPI:

domain 56*
Error Status 39, 41, 47, 53-54
extensions to ACR-NEMA 11*
Recognition Code (DE) 36*
<STATE> 19, 45*
State 9, 14
"ARCHIVED" 9*
information 9*
private 9*
"REPORTED" 9*
SPI-defined 9*
<State label> 19, 24*
SUBMIT (HLF) 13, 18*, 31
SUBMIT/SPI 31-32, 36, 44*, 57,
59-61

- T -

<TARGET> 23, 25, 27, 34
<TARGET-SUBSET> 23
<TEMPLATE> 16, 23, 24*, 49-50
<TOTAL-SIZE> 35, 54
<TYPE> 38
Type, Data Object:
Graphics 7*
Image 7*
Other 7*
Private Graphics 7*
Private Image 7*
Private Text 7*
Text 7*

- U -

UID = Identifier, Unique
<UID> 18-21, 38, 40, 42, 44-48, 51
UID element 8*
Unique Identifier 7



STANDARD PRODUCT INTERCONNECT

for

COMPATIBILITY OF DIGITAL IMAGING

NOT TO BE DISCLOSED

SPI Document 4

Data Object Formats

Document Version No. 1 Date: 1987-12-01
Document Edition No. 1 Date: 1987-12-01

This document is part of the SPI Release defined in the current "SPI Valid Documents List".

The document consists of 40 pages excluding this cover page.

Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved even in the event of the grant of a patent or the registration of a utility model or design.

CONTENTS

1.0	INTRODUCTION	2
1.1	OBJECTIVES	2
1.2	SCOPE OF THIS DOCUMENT	2
1.3	APPLICABLE DOCUMENTS	3
1.4	OTHER CONSIDERATIONS	3
2.0	CLARIFICATIONS TO ACR-NEMA DATA FORMATS	4
2.1	ENUMERATED VALUE RANGE OF ACR-NEMA DATA ELEMENTS	4
2.2	MANUFACTURER/USER SPECIFIC DATA ELEMENTS	5
2.3	IMAGE CODING AND DATA COMPRESSION	6
3.0	SPI DATA OBJECT TYPES	8
3.1	GENERAL	8
3.2	FOLDER	9
3.3	COMPRESSED IMAGE	9
4.0	DATA OBJECT IDENTIFICATION	13
4.1	UNIQUE IDENTIFIER	13
4.2	DATA OBJECT DESCRIPTION	14
4.2.1	DESC for Data Set Type Image	15
4.2.2	DESC for Data Set Type Text	16
4.2.3	DESC for Data Set Type Graphics	16
4.2.4	DESC for SPI Data Object Type Folder	16
5.0	GROUPS AND DATA ELEMENTS	17
5.1	INTRODUCTION	17
5.2	GROUP NUMBER RANGES	17
5.3	DATA ELEMENT NUMBER RANGES	17
5.4	ACR-NEMA DATA ELEMENTS	18
5.4.1	Optional ACR-NEMA Data Elements As Used by SPI	18
5.4.2	ACR-NEMA Elements with SPI-Defined Format	18
5.5	MANUFACTURER/USER (SHADOW) GROUPS	19
5.5.1	Command Information Shadow Group (0001)	19
5.5.2	Identifying Information Shadow Group (0009)	25
5.5.3	Patient Information Shadow Group (0011)	26
5.5.4	Image presentation information Shadow Group (0029)	26
5.5.5	Folder Group (0041)	27
5.5.6	Pixel Data Shadow Group (7FE1)	27
5.6	GROUP 0001 DATA ELEMENT TYPES BY COMMAND	28
5.7	ENUMERATED VALUES	30
5.8	SPI STATUS CODES	32
APPENDIX	SPI STATUS CODE VALUES	33
INDEX		38

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVES

Data Objects made up of Data Elements constitute the basic logical data structure for transferring images, commands and other data in a Picture Archiving and Communication System (PACS). In SPI, a Data Object is based on an ACR-NEMA data set in concept, but a Data Object contains ACR-NEMA-conforming extensions.

There are two major objectives for defining this SPI data structure. The first is to address several of the difficulties in implementing practical Imaging Equipment(s) and Network Interface Equipment(s) and integrating them into a PACS with the limitations of the existing ACR-NEMA Standard. The second is to provide sufficient functionality for a PACS by using SPI-defined Data Objects.

1.2 SCOPE OF THIS DOCUMENT

This document describes the formatting of all Data Objects defined and/or handled by SPI. The formatting and coding is independent of transport and storage media.

Chapter 2 describes the clarifications and extensions needed for SPI, relative to the ACR-NEMA Digital Imaging and Communications Standard.

Apart from the Data Object (data set) Types defined by ACR-NEMA, there is a need for "special" Data Object Types. A description of these Data Object Types is contained in Chapter 3.

Data Objects are uniquely identified within an SPI-based PACS. A union of this unique identification (UID) and a set of describing Data Elements (DESC) is intended to cover the elements expected to be queried for Data Object selection. Chapter 4 describes the UID scheme, which is common for all Data Object Types, and the DESC for each Data Object Type.

The actual Data Elements, their grouping, coding and semantics are described in Chapter 5.

1.3 APPLICABLE DOCUMENTS

The ACR-NEMA Standard is the following document:

"Digital Imaging and Communications Standard", ACR-NEMA Standards Publication No. 300-1985.

Country codes are taken from the following document: "International Numbering Plan for Public Data Networks", CCITT Recommendation X.121.

The SPI document set is organized as follows:

"Concepts and Requirements" are described in Document 2, "Terminology" in Document 1, "Application Services" in Document 3, "Communication Interconnect" in Document 5, and "Off-Line Media and Data Formats" in Document 6.

1.4 OTHER CONSIDERATIONS

Double quotes ("...") are normally used in the SPI documents; single quotes ('...') are used for special purposes in syntax definitions and inside text that is already quoted.

SPI ignores upper/lower case differences in ASCII strings used in command and Data Element text. The convention in other text is to capitalize commands and specific terminology for emphasis. Terms are not capitalized in general use. Thus, for example, "Data Object" and "Data Element" are capitalized, while "object" and "element" are not. A special case, "data set", remains uncapitalized (except in tables and special lists), in conformance with ACR-NEMA usage.

Dates that are not part of SPI-defined syntax are written in ISO-format, e.g., 24 September 1987 becomes 1987-09-24.

Normally in SPI documents, the convention is adopted that a hexadecimal number is designated by appending the letter "H" to the hexadecimal digit string. However, in this document all (group,element) numbers are to be understood as representing hexadecimal numbers without using the "H" postfix.

CHAPTER 2

CLARIFICATIONS
TO ACR-NEMA DATA FORMATS

2.1 ENUMERATED VALUE RANGE TO ACR-NEMA DATA ELEMENTS

Some manufacturer/user-specific information could be represented by ACR-NEMA Data Elements except for the fact that ACR-NEMA restricts these elements to a fixed set of values. Alternately, such information could be represented in a shadow group, but the ACR-NEMA Standard does not provide a meaningful value to be assigned to the corresponding standard-group Data Element in such a case.

For example, an MR spectroscopy data set might be described as follows:

(0008,0040)	DATA SET TYPE, e.g., MR spectrum
(0008,0060)	MODALITY, e.g., MRS (MR spectroscopy)
(0028,0040)	IMAGE FORMAT, e.g., circle
(0028,0060)	COMPRESSION CODE, e.g., identifier of some algorithm
(60xx,0060)	COMPRESSION CODE, e.g., identifier of some algorithm
(60xx,0110)	OVERLAY FORMAT, e.g., circle

Unfortunately, for each of these six elements, the values shown are disallowed by the ACR-NEMA Standard, since they are not members of the Enumerated Value sets for the elements. To help manufacturers and users preserve a kind of order in applications with this kind of potential problem, SPI defines an escape value for those elements that have Enumerated Value (EV) Value Type (VT):

- A 16 bit binary (BI) Value Representation (VR) shall have the escape value "EEEEH".
- ASCII Value Representation values shall have the "SPI Recognition Code" as escape value.

The meaning of an escape value is that the "real" value to be used is located in the shadow group (group+1).

This is an important distinction - SPI supports an escape mechanism for Enumerated Values although ACR-NEMA does not!

2.2 MANUFACTURER/USER-SPECIFIC DATA ELEMENTS

The ACR-NEMA Standard allows ACR-NEMA, manufacturers, and users to allocate numbers for Data Elements within a message/Data Object. No conflicts can occur between the Data Element numbers of ACR-NEMA and those of "manufacturers" and/or "users", and none can occur between the allocations of "manufacturers" and those of "users". However, the ACR-NEMA Standard does not specify how a specific manufacturer or user is to be identified, or how multiple claims on the same Data Element are to be resolved. Thus, there is no unambiguous way to recognize the Data Elements of a specific manufacturer or user, nor to avoid identification errors in a PACS where there are different identification schemes in use.

This problem is aggravated when multiple manufacturers and/or multiple users need to provide specific information within the same message/Data Object. This could happen when a Data Object is processed and displayed within a PACS which has equipment provided by multiple manufacturers, each requiring the addition of specific values. It could also happen when the Data Object needs to contain information from the same manufacturer or user in multiple forms (e.g., an old form and a new form) because of out-of-phase releases of equipment. Other cases can be imagined.

So far, ACR-NEMA has not provided a solution for this problem. Since the problem still remains in real-world PACS implementations, SPI defines a means described below to resolve the identification and potential collision problems. This technique is currently only valid in the SPI domain and cannot be used without risk in conjunction with techniques used by other manufacturers and users.

Solution:

In each odd-numbered group except the Command Information shadow group 1, a fixed block of Data Element numbers is reserved to identify the "ownership" of sets of Data Elements (each set consists of 256 elements). The elements 0010 - 007F are reserved to identify manufacturer sets of elements, while the elements 0080 - 00FF are reserved to identify user sets. The elements 1000 - 7FFF are reserved for manufacturer sets of elements and the elements 8000 - FFFF are reserved for user sets. Each identifier element (0010 - 00FF) is a type 1 free-formatted (FF) single (S) ASCII string (AT) and shall contain the "SPI Recognition Code" for blocks of Data Elements reserved by SPI.

The following reservation rules are applicable:

1. The following mapping is required:

Element	Defines	Reserved Element Block
0010		1000-10FF
007F		7F00-7FFF
0080		8000-80FF
00FF		FF00-FFFF

Manufacturer

User

2. If necessary, a manufacturer or user can take more than one set of elements using the PART or LAST extension of the identifier element (see example).
3. The manufacturer and user Data Element number ranges shall be taken in ascending order, starting from the first Data Element of the next free reserved set, which can be determined by examination of the identifier elements.
4. The effective manufacturer/user element number shall be computed as the actual manufacturer/user element number (the one that physically occurs in the Data Object) minus the base of the reserved set plus the number of preceding reserved sets for that manufacturer/user times 256:

$$\text{element number} = E\# - \text{Base} + (\text{preceding sets}) * 256$$

An illustration of the technique is given in Table 2.2-1 (page 7).

2.3 IMAGE CODING AND DATA COMPRESSION

SPI permits the use of image coding and data compression techniques that are ACR-NEMA-defined, SPI-defined, or private.

At present, there are no SPI-defined coding or compression techniques.

Table 2.2-1 Example of Contained Data Elements

DESCRIPTION		DATA ELEMENT NUMBER ALLOCATION
GROUP LENGTH	0000	Reserved Data Element numbers
IDENTIFIER ELEMENT	0010 MANUFACTURER A/PART	
IDENTIFIER ELEMENT	0011 MANUFACTURER B	
IDENTIFIER ELEMENT	0012 MANUFACTURER A/LAST	
IDENTIFIER ELEMENT	0080 USER C/LAST	
		0FFF 1000
Actual Data Element number equals effective Manufacturer A element number	1000 0000	Manufacturer A reserved set of Data Elements
		10FF 1100
Actual Data Element number equals effective Manufacturer B element number	1100 0000	Manufacturer B reserved set of Data Elements
		11FF 1200
Actual Data Element number equals effective Manufacturer A element number	1200 0100	Manufacturer A reserved set of Data Elements
		12FF 1300
		Free Manufacturer sets of data elements
		7FFF 8000
Actual Data Element number equals effective User C element number	8000 0000	User C reserved set of Data Elements
		80FF 8100
		Free User sets of data elements
		FFFF

CHAPTER 3

SPI DATA OBJECT TYPES

3.1 GENERAL

All Data Object Types defined by SPI are subtypes of the ACR-NEMA data set type "Other" if not otherwise mentioned. The subtype element (0008,0041) shall contain the SPI Recognition Code. The actual "Data Object Type" and "Subtype", if necessary, are specified in group 0009, elements 0040 and 0041. SPI does not define any value for the Data Object Subtype element (0009,0041): this element is intended for private extensions only.

Table 3.1-1 (page 10) shows the ACR-NEMA classification of groups per Data Object Type. For example, a Data Object of type Text shall include the ACR-NEMA group for Identifying Information and its shadow group (0008 and 0009) and an ACR-NEMA Text group (4000); it may optionally include the ACR-NEMA Patient Information group and its shadow group (0010 and 0011); it shall not contain any of the other ACR-NEMA groups or the special SPI-defined groups (0041 or 7FE1).

Table 3.1-2 (pages 11-12) specifies the values of the Data Set/Object Type elements for each Data Object and command combination. For example, a SEND/SPI command request sending an "Image" Data Object contains in its command group the element "Data Set Type" (0800) with the ACR-NEMA value of "Image" (0000) and contains in its command shadow group the element (0040) with the same value. This command shadow group element is patterned after the "Data Set Type" equivalent element in an ACR-NEMA Identifying group (0008,0040). The Data Object of type "Image" that is sent contains the "Image" value in its "Data Set Type" location (0008,0040) and in its "Data Set Type" shadow group (0009,0040). There is no data set part of the response message. Hence, the command response to this SEND/SPI contains "Null" (0101) in both the command group (0000,0800) and the command shadow group (0001,0040). See 5.7 for more information on Enumerated Values of ACR-NEMA and SPI Data Elements.

3.2 FOLDER

The purpose of the Folder Data Object Type is to allow SPI PACS users to refer collectively to arbitrarily-chosen groups of Data Objects that are considered meaningful.

A Folder consists of identifying information, a free-form textual element that describes the purpose of the Folder (for user interpretation), an Enumerated Value element describing the characteristic property of the Folder (for PACS interpretation), and a list of one or more Unique Identifiers of Data Objects (which may themselves be Folders).

A given Data Object may be referred to in zero or more Folders.

3.3 COMPRESSED IMAGE

The purpose of the Compressed Image Data Object Type is to allow handling of compressed images within the SPI domain.

A Compressed Image is a subtype of the ACR-NEMA data set "Private Image" and contains the ASCII escape value (see 2.1) in the Compression Code (element (0028,0060)), an identification of the used compression algorithm (element (0029,0060)) and the Compressed Pixel Data (element (7FE1,0010)).

Table 3.1-1 Classification of Groups for Each Data Object Type

Data Object Type (Abbreviation)	Groups									
	0008 0009	0010 0011	0018	0020	0028 0029	0041	4000	6000 601E	7FE0	7FE1
Image	R R	R O	R	R	R N	N	O	O	R	N
Compressed Image (Compr Image)	R R	R O	R	R	R R	N	O	O	N	R
Graphics (Graph)	R R	O O	O	O	N N	N	O	*	N	N
Text	R R	O O	N	N	N N	N	R	N	N	N
Folder	R R	O O	O	O	N N	R	N	N	N	N
Identifier (Ident)	R R	O O	O	O	O O	N	N	N	N	N
Null	N N	N N	N	N	N N	N	N	N	N	N

R = Required

O = Optional

N = Not allowed

* = At least one of the groups is required;
the other groups are optional.

Table 3.1-2 Data Object Type Elements and Command Relation (Part 1)

	Data Object Element	Image	Compr* Image	Graph*	Text	Folder
	0008,0040 0008,0041	Image —	Priv* Image **	Graph —	Text —	Other **
Command Request	0009,0040 0009,0041	Image —	Compr Image —	Graph —	Text —	Folder —
SEND/SPI	0000,0800 0001,0040	Image Image	Priv Image Compr Image	Graph Graph	Text Text	Other Folder
READY/SPI	0000,0800 0001,0040	Ident* Ident	Ident Ident	Ident Ident	Ident Ident	Ident Ident
QUERY/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
MOVE/SPI	0000,0800 0001,0040	Ident Ident	Ident Ident	Ident Ident	Ident Ident	Ident Ident
COPY/SPI	0000,0800 0001,0040	Ident Ident	Ident Ident	Ident Ident	Ident Ident	Ident Ident
GCBEGIN/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
GCEND/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
SUBMIT/SPI	0000,0800 0001,0040	Ident Ident	Ident Ident	Ident Ident	Ident Ident	Ident Ident
CHANGESTATE/SPI	0000,0800 0001,0040	Ident Ident	Ident Ident	Ident Ident	Ident Ident	Ident Ident
DESUBMIT/SPI	0000,0800 0001,0040	Ident Ident	Ident Ident	Ident Ident	Ident Ident	Ident Ident
FORGET/SPI	0000,0800 0001,0040	Ident Ident	Ident Ident	Ident Ident	Ident Ident	Ident Ident
CONFIGURE/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null

* Abbreviations for most Data Object Types are given in Table 3.1-1
(page 10). "Priv Image" means "Private Image".

** = "SPI Release 1"

Table 3.1-2 Data Object Type Elements and Command Relation (Part 2, concluded)

	Data Object Element	Image	Compr* Image	Graph*	Text	Folder
	0008,0040 0008,0041	Image —	Priv* Image **	Graph —	Text —	Other **
Command Response	0009,0040 0009,0041	Image —	Compr Image —	Graph —	Text —	Folder —
SEND/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
READY/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
QUERY/SPI match responses	0000,0800 0001,0040	Ident* Ident	Ident Ident	Ident Ident	Ident Ident	Ident Ident
other responses	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
MOVE/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
COPY/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
GCBEGIN/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
GCEND/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
SUBMIT/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
CHANGESTATE/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
DESUBMIT/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
FORGET/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null
CONFIGURE/SPI	0000,0800 0001,0040	Null Null	Null Null	Null Null	Null Null	Null Null

* Abbreviations for most Data Object Types are given in Table 3.1-1 (page 10). "Priv Image" means "Private Image".

** = "SPI Release 1"

CHAPTER 4

DATA OBJECT IDENTIFICATION

The set of Data Elements that together constitute a given Data Object is divided by SPI into three disjoint subsets:

- Unique Identifier (UID)
With this set (which contains only one element), it is possible to uniquely distinguish each Data Object from every different Data Object.
- Data Object Description (DESC)
This set enables a user to query for Data Object selection (see Document 3).
- The remaining elements.

4.1 UNIQUE IDENTIFIER

The ACR-NEMA Standard does not define a specific means for unique data set identification, although it does indicate certain possibilities for data sets of type "Image". To assure integrity of operations in an SPI-based PACS, SPI requires that each Data Object shall be uniquely distinguished from every different object. Since the ACR-NEMA Standard does not assure this except via bitwise comparison, SPI defines the Unique Identifier element (0009,0015) for this purpose. To guarantee that an SPI-conforming Data Object can be uniquely identified within any SPI-based PACS, an SPI Unique Identifier is defined as a 26 character ASCII string having the following layout:

UID = <PACS#><IE#><DO#>

PACS# : A six character ASCII string specifying a world-wide unique PACS identification:

- Three characters indicating the Data Country Code (DCC) as defined in the CCITT recommendation X.121.
- Three characters uniquely identifying the PACS within the DCC, consisting of a manufacturer abbreviation and a serial number.

IE# : A four character ASCII string uniquely identifying the originating modality within <PACS#>. One possibility for a unique IE# is:

- Two characters indicating the modality type (as specified by ACR-NEMA).
- Two characters indicating the modality number.

DO# : A 16-character ASCII string unique within <PACS#><IE#>. A suggested form is a union of the ACR-NEMA specified date and time, with the following exceptions:

- The separators, "." (dot) and ":" (colon), are excluded.
- The second fraction consists of only two characters.

Example: PACS# = 342Z01 (PACS 01 of manufacturer "Z" in Barbados)

IE# = MR02 (MR unit number 2)

DO# = 1986070308351234 (July 3, 1986, 8:35:12.34 AM)

UID = 342Z01MR021986070308351234

4.2 DATA OBJECT DESCRIPTION

The purpose of the Data Object Description (DESC) is to have, per Data Object Type, a set of Data Elements that allows most queries for Data Object selection to be answered. The DESC elements are not always mandatory elements, but when available shall be supplied and subsequently recorded by IMS when a Data Object is "submitted". Apart from the DESC, a user can also provide other elements, which, depending on the implementation, IMS may record. These other elements shall belong to groups 08...29.

There are two classes of DESCs, a "fixed" one for each ACR-NEMA and Data Object Type, and a "free-formatted" one for private extended Data Object Types.

A "fixed" DESC consists of a set of defined Data Elements. The following sections describe the fixed elements for each ACR-NEMA and Data Object Type.

A "free-formatted" DESC enables the user to specify those elements which are to be mandatorily retained by IMS to support queries for Data Objects of types not defined in SPI. An IMS must support at least 10 elements, with a minimum total length of 512 bytes for all elements together. The DESC elements shall include the elements Data Set Type (0008,0040), Data Set Subtype (0008,0041), Data Object Type (0009,0040) and Data Object Subtype (0009,0041); the "Comments" element (0009,0010) is recommended, but optional. The "free-formatted" DESC description (element 0009,0016) consists of an even number of 16-bit binary (ACR-NEMA BI type) values. Each pair of consecutive 16-bit binary values indicates corresponding group and element numbers; these must be in ascending order.

4.2.1 DESC for Data Set Type Image

Group	Element	Definition
0008	0020	Study Date
	0030	Study Time
	0040	Data Set Type
	0041	Data Set Subtype
	0060	Modality
	0090	Referring Physician
	1010	Station Identification
	1030	Procedure Description
	1040	Institutional Department
	1050	Attending Physician
	1060	Radiologist
	1080	Admitting Diagnosis
	4000	Comments
0009	0010	Comments
	0040	Data Object Type
	0041	Data Object Subtype
	00A0	Resulting Diagnosis
0010	0010	Patient Name
	0020	Patient Identification
	0030	Patient Birthdate
	0040	Patient Sex
	1005	Patient Maiden Name
	1010	Patient Age
	0010	Organ
0011	0015	Allergy Indication
	0020	Pregnancy
0020	0010	Study
	0011	Series
	0012	Acquisition
	0013	Image
	3401	Modifying Device Identification
	3402	Modified Image Identification
	3403	Modified Image Date
0028	3405	Modified Image Time
	0010	Rows
	0011	Columns
	0101	Bits Stored

4.2.2 DESC for Data Set Type Text

Group	Element	Definition
0008	0020	Study Date
	0030	Study Time
	0040	Data Set Type
	0041	Data Set Subtype
	1030	Procedure Description
	1060	Radiologist
	1080	Admitting Diagnosis
0009	4000	Comments
	0010	Comments
	0040	Data Object Type
	0041	Data Object Subtype
	00A0	Resulting Diagnosis
	00A1	Report Date
0010	00A2	Report Time
	0010	Patient Name
	0020	Patient Identification
	0030	Patient Birthdate
	0040	Patient Sex
	1005	Patient Maiden Name
0011	1010	Patient Age
	0010	Organ

4.2.3 DESC for Data Set Type Graphics

Group	Element	Definition
0008	0020	Study Date
	0030	Study Time
	0040	Data Set Type
	0041	Data Set Subtype
	1010	Station Identification
0009	4000	Comments
	0010	Comments
	0040	Data Object Type
	0041	Data Object Subtype
0020	0010	Study
	0011	Series
	0012	Acquisition
	0013	Image

4.2.4 DESC for Data Object Type Folder

Group	Element	Definition
0008	0040	Data Set Type
	0041	Data Set Subtype
0009	0010	Comments
	0040	Data Object Type
	0041	Data Object Subtype

CHAPTER 5

GROUPS AND DATA ELEMENTS

5.1 INTRODUCTION

This chapter defines all Data Elements and their grouping both for ACR-NEMA and for manufacturer or user purposes. It contains only SPI additions and extensions to the ACR-NEMA Standard. Parts needing no additions are empty. The descriptions of the Data Elements are organized according to the ACR-NEMA list of elements. Each group of related Data Elements is a section. Reserved group and Data Element number ranges for manufacturers or users are indicated.

See the ACR-NEMA Standard for definitions of Value Representations (VR), Value Types (VT), Value Multiplicity (VM) and Data Element Types.

5.2 GROUP NUMBER RANGES

Reserved by ACR-NEMA	: Even numbers in 0000...FFFE
Reserved for shadow groups	: ACR-NEMA standard group numbers+1
Reserved for manufacturers and users	: Other odd numbers

5.3 DATA ELEMENT NUMBER RANGES

For all odd-numbered groups, except the Command Information shadow group (0001), the following SPI rules apply (see 2.2):

- Each group shall contain mandatory Data Elements for the manufacturer and/or user to identify reserved sets of Data Elements.
- The Data Element numbers 0000...00FF are reserved for group length and the above-described identifiers.
- Blocks of Data Elements reserved by SPI shall be identified by the "SPI Recognition Code" as defined in 5.7.
- Data element numbers 0100...0FFF are reserved for ACR-NEMA extensions.
- Data element numbers 1000...7FFF are reserved for the manufacturer(s) identified by the manufacturer recognition code(s).
- Data element numbers 8000...FFFF are reserved for the user(s) identified by the user recognition code(s).
- Specific manufacturer/user Data Elements are identifiable by computation with the base of the reserved sets of Data Elements.

5.4 ACR-NEMA DATA ELEMENTS

5.4.1 Optional ACR-NEMA Data Elements As Used by SPI

SPI uses the optional ACR-NEMA Data Elements listed in Table 5.4-1.

Table 5.4-1 SPI Uses of Optional ACR-NEMA Data Elements

Group	Element	Name	Type NEMA	Type SPI	Comment
0008	0041	Data Set Subtype	3	2	see 4.1
0020	0011	Series	2D	2	
0020	0012	Acquisition	2D	2	
0020	0013	Image	2D	2	

5.4.2 ACR-NEMA Elements with SPI-Defined Format

The ACR-NEMA Standard defines the Logical Addresses as free format. A PACS network has an internal structure, e.g., there might be subnetworks, etc. Therefore, SPI defines the following format for those ACR-NEMA elements specifying Logical Addresses:

PACS-IE Ident = <PACS#><IE#><NN#>

PACS# : Same as PACS# defined in 4.1

IE# : Same as IE# defined in 4.1

NN# : A four character ASCII string uniquely identifying the network node within <PACS#>:

- Two characters indicating the network number.
- Two characters indicating the PACS node number.

This PACS-IE Ident format is mandatory within the SPI domain for the following ACR-NEMA elements:

0000,0200
0000,0300
0000,0400
0000,0600

5.5 MANUFACTURER/USER (SHADOW) GROUPS

The "actual" Command Information shadow group elements and the "effective" other SPI Data Elements are defined in the following subsections. The elements "Group Length" and "Identifier" are not included (see 2.2) in the description, but they shall be present.

5.5.1 Command Information Shadow Group (0001)

Table 5.5-1 Command Information Shadow Group Elements

Group	Actual Element Number	Name	Type	VR	VT	VM	Comment
							see note 1
0001	0010	SPI Recog. Code	1	AT	EV	S	
0001	0018	SPI Command	1	BI	EV	S	see note 2
0001	0019	SPI Status Code	see 5.6	BD	EV	M	see 5.8
0001	001A	Status Comment	see 5.6	AT	FF	S	see 5.8
0001	0020	IMS-FLAG	see 5.6	AT	EV	S	
0001	0021	STATE	see 5.6	AT	DF	M	see note 3
0001	0022	FORGET-LAST-COPY	see 5.6	AT	EV	S	
0001	0028	SESSION-ID	see 5.6	BI	HX	S	
0001	0029	COMPLETION-STATE	see 5.6	AT	EV	S	
0001	0030	CLASS	see 5.6	AT	EV	S	
0001	0040	Data Object Type	1D	BI	EV	S	see 5.7
0001	0050	Location List	see 5.6	AT	DF	M	see note 4
0001	0051	Forgotten Status	see 5.6	BD	DF	M	see note 5
0001	0060	Wait Time	see 5.6	BI	HX	S	see note 6
0001	0070	Service	see 5.6	AT	EV	S	
0001	0071	Move Dest Service	see 5.6	AT	EV	S	
0001	0080	DATA-OBJECT-SIZE	see 5.6	BD	HX	S	see note 7
0001	0081	NUM-OBJECTS	see 5.6	BI	HX	S	see note 8
0001	0090	Query Template	see 5.6	AT	DF	M	see note 9
0001	0091	MAX-RESPONSES	see 5.6	BI	HX	S	
0001	0092	RETURN-INFO-FLAG	see 5.6	AT	EV	S	
0001	0093	COUNT	see 5.6	AT	EV	S	

Table 5.5-1 Notes

1. The Command Information shadow group is the only Manufacturer/User group that has ACTUAL Element Numbers.

The Enumerated Values are defined in 5.7.

2. In 5.7, the related ACR-NEMA command field values (element 0000,0100) are given.
3. The defined format for the STATE element is: <State label> '=' <value>. SPI-defined <State label>s are "ARCHIVED" and "REPORTED", while SPI-defined <value>s are "YES" and "NO". See Document 3 for more detailed information.
4. The purpose of the Location List element is for use in the QUERY/SPI and FORGET/SPI commands.

In the QUERY/SPI command, it indicates that:

- No match-responses of a QUERY/SPI-REQUEST shall be returned that do not mention at least one of the location(s) specified. The element shall specify in the request either a list of specific locations or "ALL". The response shall specify the list of locations that contain a copy of the matching Data Object.
- Matches are not limited to Data Objects. When not limited, the element shall specify "NONE" for the request as well as for the response.

For the FORGET/SPI command:

- The element shall specify in the request either a list of specific locations or "ALL". The response shall specify the location(s) at which a Data Object is to be forgotten or has been forgotten.

So the element specifies one of these possibilities:

- A list of one or more IEs whereby each location is identified by its PACS-IE Ident (see 5.4.2).
- The ASCII text string "ALL" (for requests only).
- The ASCII text string "NONE" (for the QUERY/SPI command only).

5. The Forgotten Status element specifies the result for each attempt to FORGET an object. Its format is yes/no indication (yes = "0000" no = "0001") followed by one of the Severity Status Codes as defined in 5.8. This element shall have the same number of values as there are locations in the Location List element; each status value corresponds to the respective entry in the location list.
6. When an SPI Status Code indicates "WAIT", the Wait Time element specifies the wait time (in seconds) before the request can be retransmitted.
7. The **DATA-OBJECT-SIZE** element specifies the size (in bytes) of the object to be transmitted for a READY/SPI command or the estimated total size (in bytes) of the group of objects to be transmitted in a GROUPCOPY session (parameter of the GCBEGIN/SPI-REQUEST command). (Refer to the use of "<TOTAL-SIZE>" in 4.3 of Document 3.)

8. The NUM-OBJECTS element contains the number of objects to be transmitted within a GROUPCOPY session (parameter of the GCBEGIN/SPI-REQUEST command) or the number of objects that can be received by the destination when the SPI Status Code indicates "TOO-MANY-OBJECTS". The default value for this element is "zero".
9. The Query Template element specifies the template to be used for a query. The ShadowOwnerCode for SPI groups is the "SPI Recognition Code" defined in Section 5.7. See Document 3 for more details.

Example of a Query Template

([0008,0020]<="1984.07.18")OR([0009,"SPI Release 1",00A1]>="1985.10.12")

Table 5.5-2 specifies the Command Information shadow group elements that are valid only for CONFIGURE/SPI-REQUESTS and -RESPONSES.

Table 5.5-2 Command Information Shadow Group Elements Valid for CONFIGURE/SPI-REQUESTS and -RESPONSES

Group	Actual Element Number	Name	Type	VR	VT	VM	Comment
0001	0100	PACS-IE Ident	1	AT	DF	S	see 5.4.2
0001	0102	Mod. Type	1	AT	EV	S	
0001	0104	Mod. Type Exten.	3	AT	FF	S	
0001	0106	Station ID	1	AT	FF	S	
0001	0108	Inst. Department	3	AT	FF	S	
0001	010A	Coarse Capabil.	*	AT	DF	M	see note 1
0001	010C	Fine Capabil.	*	AT	DF	S	see note 1
0001	010E	RESPONSES-FLAG	*	AT	EV	S	see note 2
0001	0110	Format	*	AT	DF	M	see note 3
0001	0112	Query Elements	*	AT	DF	M	see note 4
0001	0114	Query Operators	*	AT	EV	M	see note 5
0001	0116	Query Match Ret.	*	AT	EV	S	see note 6

Table 5.5-2 Notes

1. Each capability is identified by a "Composite Capability Identification" (CCI) consisting of the capability name followed by a list of one or more command names with their properties.

Rules:

- All CCI elements are ASCII text strings (see ASCII values in the ACR-NEMA Standard).
The control characters "carriage return", "line feed" and "tab" are ignored.
- CCIs are separated by the backslash "\" character.
- Individual CCI elements are separated by the plus "+" character.
- A CCI shall not be specified when the capability (Service) is not available in the modality.
- Each capability name consists of a Service name (see element (0001,0070)) followed by one or more CLASSES (see element (0001,0030)) embedded in square brackets "[" and "]" with the format:
" [<Service>+<CLASS-1>+ ... +<CLASS-n>]".
If the Service does not have any CLASS available, the capability name contains an empty CLASS (" [<Service>+]").
- Each capability name is followed by one or more command names. If a particular command name is specified more than once for the same capability, the meaning is unspecified.
- Each command name has the format "<command>" where <command> is:

SUBMIT/SPI	GCBEGIN/SPI	SEND
CHANGESTATE/SPI	GCEND/SPI	MOVE
DESUBMIT/SPI	READY/SPI	GET
FORGET/SPI	SEND/SPI	FIND
COPY/SPI	MOVE/SPI	ECHO
QUERY/SPI		CANCEL
CONFIGURE/SPI		DIALOG
- Each command name is followed by one or more properties. If a particular property is specified more than once for the same command, the meaning is unspecified. If a property is mentioned, the property shall be available, otherwise it is not. Properties are:
 - a. FINE
Indication that the modality is able to respond to a "FINE" CONFIGURE/SPI-REQUEST for this command.
 - b. ISSUE
Indication that the modality can "issue" the command.
 - c. ACCEPT
Indication that the modality can "accept" the command and perform the requested function.

The Coarse Capabilities element (0001,010A) contains the capabilities of the requesting IE when issuing a CONFIGURE/SPI-REQUEST or the capabilities of the responding IE when that IE responds to a request with the RESPONSES-FLAG element (0001,010E) set to "COARSE" or "FINE".

Example of a Coarse Capability Element (0001,010A) with 2 CCIs

```
[PRS+]
:READY/SPI+ACCEPT+ISSUE
:SEND/SPI+ACCEPT+ISSUE
:MOVE/SPI+ACCEPT
:SEND+ACCEPT+ISSUE
:MOVE+ACCEPT\
[PBS+TEMP]
:SUBMIT/SPI+ISSUE
:FORGET/SPI+ISSUE
:GCBEGIN/SPI+ACCEPT+ISSUE
:GCEND/SPI+ACCEPT+ISSUE
:READY/SPI+ACCEPT+ISSUE
:SEND/SPI+ACCEPT+ISSUE
:MOVE/SPI+ACCEPT+ISSUE
:SEND+ACCEPT+ISSUE
:MOVE+ACCEPT+ISSUE
```

The capability name of a CCI for a Fine Capability shall only contain one Service and one command name: properties are not required.

Examples of Fine Capability Element (0001,010C)

```
[IMS+PERM]
:QUERY/SPI
```

```
[EXS+PRINT]
:GCBEGIN/SPI
```

2. When the RESPONSES-FLAG element specifies "FINE", the Fine Capability element (0001,010C) is mandatory.
3. This element specifies the (PRINT) formats the Export Service can handle. The element is mandatory in the response to a "FINE" CONFIGURE/SPI-REQUEST for the CLASS "PRINT". No formats are defined by SPI at present.

4. This element specifies the list of the Data Element number pairs (group,element) with which the database can be queried. The values in the list are separated by the backslash "\" character. The element is mandatory in the response to a "FINE" CONFIGURE/SPI-REQUEST for the command QUERY/SPI.

The format of the element is: <Group> ',' <Element>

<Group> ::= <EvenInteger>
| <OddInteger> ',' <ShadowOwnerCode>

<ShadowOwnerCode> ::= <ASCIITextString>

<Element> ::= <Integer>

<Integer> and <ASCIITextString> are as defined by the ACR-NEMA Standard for the value representations BI and AT, respectively. <EvenInteger> represents an even <Integer> number, while <OddInteger> represents an odd <Integer> number. Within the ASCII Text String, the special characters '*' and '@' may be used with the meanings defined by the ACR-NEMA Standard. The value 'NULL' is used as a wildcard for integer values, having a meaning equivalent to '*' as used for strings. The <ShadowOwnerCode> for SPI groups is the "SPI Recognition Code" defined in Section 5.7.

Example:

0008,0020\0009,"SPI Release 1",00A1

5. This element specifies the set of operators that can be used for a QUERY/SPI-REQUEST. The element is mandatory in the response to a "FINE" CONFIGURE/SPI-REQUEST for the command QUERY/SPI.
6. This element specifies whether the IE is able to return the number of objects which match the criteria. The element is mandatory in the response to a "FINE" CONFIGURE/SPI-REQUEST for the command QUERY/SPI.

5.5.2 Identifying Information Shadow Group (0009)

Table 5.5-3 Identifying Information Shadow Group Elements

Group	Effective Element Number	Name	Type	VR	VT	VM	Comment
0009	0010	Comments	3	AT	FF	S	see note 1
0009	0015	Unique Identifier	1	AT	DF	S	see 4.1
0009	0016	DESC Descriptor	3	BI	DF	M	see 4.2
0009	0017	Special Ident.	3	AT	FF	S	Manufact. specific
0009	0040	Data Object Type	1D	BI	EV	S	see 5.7
0009	0041	Data Object Subt.	3	AT	FF	S	
0009	0061	Mod. Type Exten.	3	AT	FF	S	
0009	00A0	Resulting Diag.	3	AT	FF	S	
0009	00A1	Report Date	3	AT	DF	S	NEMA
0009	00A2	Report Time	3	AT	DF	S	NEMA
0009	00C0 +2*(i-1)	Text relating to a modified object [i > 0]	3	AT	FF	S	see note 2
0009	00C1 +2*(i-1)	Prev. UID of the modified object [i > 0]	3	AT	DF	S	see note 2 and 4.1

Table 5.5-3 Notes

1. A user can add a comment to a Data Object using the element 0010. Since this element is part of the "fixed" DESCs and recommended for the "free-formatted" DESCs (see 4.2), it is advisable to specify the general commentary for a Data Object (e.g., a Folder) in this element.
2. The elements 00C0 and 00C1 give the modification history of the object (see also Document 3). The value of "i" is the number of modifications that have been made. For example, suppose two modifications have occurred. Then element 00C0 contains text describing the purpose of the first modification, and 00C1 contains the original object's UID. Element 00C2 contains text describing the purpose of the second modification, and 00C3 contains the UID of the object with the first modification.

5.5.3 Patient Information Shadow Group (0011)

Table 5.5-4 Patient Information Shadow Group Elements

Group	Effective Element Number	Name	Type	VR	VT	VM	Comment
0011	0010	Organ	3	AT	FF	M	
0011	0015	Allergy Indic.	3	AT	FF	M	
0011	0020	Pregnancy	3	AT	EV	S	

5.5.4 Image Presentation Information Shadow Group (0029)

Table 5.5-5 Image Presentation Information Shadow Group Elements

Group	Effective Element Number	Name	Type	VR	VT	VM	Comment
0029	0060	Compression Algorithm	1	AT	FF	S	

5.5.5 Folder Group (0041)

Table 5.5-6 Folder Group Elements

Group	Effective Element Number	Name	Type	VR	VT	VM	Comment
0041	0010	Folder Type	1	BI	EV	S	
0041	0020	UID for Object #1	1	AT	DF	S	see 4.1
0041	0021	Text Relating to Object #1	3	AT	FF	S	
0041	0022	UID for Object #2	3	AT	DF	S	see 4.1
0041	0023	Text Relating to Object #2	3	AT	FF	S	
0041	0020 +2*(i-1)	UID for Object #i [i > 1]	3	AT	DF	S	see 4.1
0041	0021 +2*(i-1)	Text Relating to Object #i [i > 1]	3	AT	FF	S	

5.5.6 Pixel Data Shadow Group (7FE1)

Table 5.5-7 Pixel Data Shadow Group Elements

Group	Effective Element Number	Name	Type	VR	VT	VM	Comment
7FE1	0010	Compressed Pixel Data	1	BI	DF	S	

5.6 GROUP 0001 DATA ELEMENT TYPES BY COMMAND

Table 5.6-1 Command Information Shadow Group Data Element Types
by Command (Part 1)

Command Request	Shadow Group Data Element Number																		
	19	1A	20	21	22	28	29	30	50	51	60	70	71	80	81	90	91	92	93
SUBMIT/SPI													1						
CHANGESTATE/SPI				1									1						
DESUBMIT/SPI				1D									1						
FORGET/SPI									1				1						
COPY/SPI			1			3		1					1	1					
QUERY/SPI									1				3			1	1	1	1
CONFIGURE/SPI													1						
GCBEGIN/SPI			1					1					1		1	1			
GCEND/SPI						1	2D						1						
READY/SPI			1			*		1					1		1				
SEND/SPI			1			*		1					1						
MOVE/SPI			1			*		1					1	1					

* - Type is mandatory (1) if these commands are part of a GROUPCOPY session.

Table 5.6-1 Command Information Shadow Group Data Element Types
by Command (Part 2, concluded)

Command Response	Shadow Group Data Element Number																		
	19	1A	20	21	22	28	29	30	50	51	60	70	71	80	81	90	91	92	93
SUBMIT/SPI	1	3																	
CHANGESTATE/SPI	1	3																	
DESUBMIT/SPI	1	3																	
FORGET/SPI	1	3								1	1								
COPY/SPI	1	3																	
QUERY/SPI	1	3		*						**		3							
CONFIGURE/SPI	1	3																	
GCBEGIN/SPI	1	3				1					1				1D				
GCEND/SPI	1	3																	
READY/SPI	1	3									1								
SEND/SPI	1	3																	
MOVE/SPI	1	3																	

* - The STATE element shall be specified (mandatory (1)) when matches are returned.

** - The Location List element shall be specified (mandatory (1)) when match-responses are returned and the element did not specify "NONE" at the QUERY/SPI-REQUEST.

5.7 ENUMERATED VALUES

SPI defines the Enumerated Values in Table 5.7-1. These extend the set of values given in Section 10 of the ACR-NEMA Standard. In Table 5.7-1, default ASCII values are underlined.

Table 5.7-1 SPI Enumerated Values (Part 1)

Data Element	Name	Values
0001,0010	SPI Recog. Code	"SPI Release 1"
0001,0018	SPI Command	<p>Related ACR-NEMA command field value (0000,0100)</p> <p>0001 = SEND/SPI-REQUEST SEND-RQ 8001 = SEND/SPI-RESPONSE SEND-RSP 0008 = READY/SPI-REQUEST SEND-RQ 8008 = READY/SPI-RESPONSE SEND-RSP 0020 = QUERY/SPI-REQUEST FIND-RQ 8020 = QUERY/SPI-RESPONSE FIND-RSP 0021 = MOVE/SPI-REQUEST MOVE-RQ 8021 = MOVE/SPI-RESPONSE MOVE-RSP 0022 = COPY/SPI-REQUEST MOVE-RQ 8022 = COPY/SPI-RESPONSE MOVE-RSP 0028 = GCBEGIN/SPI-REQUEST SEND-RQ 8028 = GCBEGIN/SPI-RESPONSE SEND-RSP 0029 = GCEND/SPI-REQUEST SEND-RQ 8029 = GCEND/SPI-RESPONSE SEND-RSP 0040 = SUBMIT/SPI-REQUEST SEND-RQ 8040 = SUBMIT/SPI-RESPONSE SEND-RSP 0041 = CHANGESTATE/SPI-REQUEST SEND-RQ 8041 = CHANGESTATE/SPI-RESPONSE SEND-RSP 0048 = DESUBMIT/SPI-REQUEST SEND-RQ 8048 = DESUBMIT/SPI-RESPONSE SEND-RSP 0050 = FORGET/SPI-REQUEST SEND-RQ 8050 = FORGET/SPI-RESPONSE SEND-RSP 00F0 = CONFIGURE/SPI-REQUEST SEND-RQ 80F0 = CONFIGURE/SPI-RESPONSE SEND-RSP</p>
0001,0019	SPI Status Code	See 5.8
0001,0020	IMS-FLAG	"SUBM" "NOTKNOWN"
0001,0022	FORGET-LAST-COPY	"YES" " <u>NO</u> "
0001,0029	COMPLETION-STATE	" <u>GO ON</u> " "CANCEL"
0001,0030	CLASS	"TEMP" "PERM" "PRINT" "EXCH"

Table 5.7-1 SPI Enumerated Values (Part 2, concluded)

Data Element	Name	Values
0001,0040	Data Object Type	<p>All ACR-NEMA-defined Data Set Type values plus the following list</p> <p>0001 = Compressed Image 0010 = Folder</p> <p>The default value is the value of the ACR-NEMA Data Set Type element (0000,0800)</p>
0001,0070	Service	"EXS" "PRS" "PBS" "IMS" "CFS"
0001,0071	Move Dest Service	See element 0001,0070
0001,0092	RETURN-INFO-FLAG	"TEMPLATE" "DESC" "ALL"
0001,0093	COUNT	"YES" "NO"
0001,0102	Mod. Type	See ACR-NEMA element 0008,0060
0001,010E	RESPONSES-FLAG	" <u>NONE</u> " "COARSE" "FINE"
0001,0111	Query Operators	"=" "<>" ">" ">=" "<" "<=" "NOT" "AND" "OR"
0001,0116	Query Match Ret.	"YES" "NO"
0009,0040	Data Object Type	<p>See element 0001,0040</p> <p>The default value is the value of the ACR-NEMA Data Set Type element (0008,0040)</p>
0011,0020	Pregnancy	"YES" "NO"
0041,0010	Folder Type	<p>0001 = Data Exchange 0002 = Teaching Case 0003 = Hard Copy 0004 = History 0005 = Case 0006 = Patient 0007 = Research</p>

5.8 SPI Status Codes

When the ACR-NEMA Status element (0000,0900) contains the value AF00H, the actual SPI status is specified in element (0001,0019). This SPI Status Code consists of a 32-bit value that is interpreted as two 16-bit unsigned integers. The first indicates the Severity Status; the second gives the Error Status within the Severity Status. The default value for the Error Status is "0000". The SPI status is a multiple value element. The list of values contains all received responses within one transaction in received order; i.e., the first received response will be the first one in the list, and the last received response will be the last one in the list.

Extra information about the SPI status or a text message for an operator can be specified in element (0001,001A). Such information could be useful for human interpretation of a situation.

Severity Status Codes (see Table A-1 (pages 33-34) and Error Status Codes (see Table A-2 (pages 35-37) are defined in the Appendix. The table for the Error Status Codes contains, when applicable, a sub-table for each Severity Status Code. An example of SPI Status Code responses to commands is included below.

Example

<u>Commands</u>	<u>SPI Status Code</u>
MOVE/SPI-REQUEST	
READY/SPI-REQUEST	
READY/SPI-RESPONSE	0000 0000
SEND/SPI-REQUEST	
SUBMIT/SPI-REQUEST	
SUBMIT/SPI-RESPONSE	0000 0000 \ A600 0000
SEND/SPI-RESPONSE	0000 0000 \ A600 0000 \ A900 0010
MOVE/SPI-RESPONSE	0000 0000 \ A600 0000 \ A900 0010 \ A900 0000

See the Appendix for meanings of the SPI Status Codes.

APPENDIX

SPI STATUS CODE VALUES

Table A-1 Severity Status Codes (Part 1)

<u>Name</u>	<u>Code</u>	<u>Description</u>
SUCCESS	0000	The command request receiver properly completed the function implied by the command request.
PENDING	FF00	The command request receiver is processing the function implied by the command request.
CANCELLED	FE00	A CANCEL-REQUEST was received and the function is "rolled back".
INTERRUPTED	FD00	The command request receiver stopped the completion of the function in a defined regular way.
CAP-NOT-AVAIL	A000	The receiver does not support the command, or the receiver does not have the ability to support the command request at this time.
INFO-NOT-AVAIL	A100	The information is not available at the receiver. The receiver or IMS has no knowledge of a Data Object with the specified <UID>.
INFO-NOT-SPEC	A200	The information requested is not uniquely specified. The receiver does not have enough information to complete the function implied by the command request.
UNKNOWN-DEST	A300	The MOVE-DESTINATION is not known to the receiver.
COMPR-NOT-SUPP	A400	The receiver does not support image compression.
NO-DIALOG	A500	No DIALOG receiver is available.
WAIT	A600	The command request receiver is currently unable to perform the command request, but expects to be able to perform it soon.

Table A-1 Severity Status Codes (Part 2, concluded)

<u>Name</u>	<u>Code</u>	<u>Description</u>
ALREADY-KNOWN	A700	The information already exists at the receiver.
RULE-VIOLATION	A800	The receiver cannot perform the request, because the initiator did not obey the rules for this request.
SUBSEQUENT-ERROR	A900	The receiver cannot perform the request; the problems are outside of the receiver.
REFUSED	AA00	The receiver will not perform the request, because it would violate the receiver's rules.
PROT-ERROR	AB00	The receiver did not expect this command request (protocol error).
SYNTAX-ERROR	C300	The command violated the SPI-defined syntax.
COMM-REJ	C400	The receiver could not receive the complete message (message size was too big).
NO-SPI-STRUC	C500	The command request did not have the SPI format structure for group 0001.
OTHER	CFxx	No specified standard reason; i.e., none of the above.

Table A-2 Error Status Codes (Part 1)

----- Severity Status = INTERRUPTED (FD00) -----		
<u>Name</u>	<u>Code</u>	<u>Description</u>
MATCH-OVERFLOW	0002	The receiver already returned <MAX-RESPONSES> matches to a QUERY/SPI.
----- Severity Status = CAP-NOT-AVAIL (A000) -----		
<u>Name</u>	<u>Code</u>	<u>Description</u>
CAP-NOT-IMPLEM	0010	The requested capability is not implemented.
NO-FINE-INFO	0030	No Fine Capability information for the requested capability is available.
SERV-NOT-IMPLEM	0110	The specified Service is not implemented.
SERV-NOT-AVAIL	0210	The specified Service is not available.
S/C-NOT-IMPLEM	0130	The specified Service/CLASS combination is not implemented.
S/C-NOT-AVAIL	0230	The specified Service/CLASS combination is not available.
S/C/F-NOT-IMPLEM	0170	The specified Service/CLASS/IMS-FLAG combination is not implemented.
S/C/F-NOT-AVAIL	0270	The specified Service/CLASS/IMS-FLAG combination is not available.
MEDIUM-NOT-AVAIL	0E10	The specified medium is not available.
FORMAT-NOT-AVAIL	0E20	The specified Format is not available.
SIZE-TOO-BIG	0A00	The specified size cannot be accepted now.
TOO-MANY-OBJECTS	0A10	The specified NUM-OBJECTS cannot be accepted now.
STATE-OVERFLOW	0B10	The CHANGESTATE/SPI results in an overflow for the non-SPI-defined State storage area.
TEMPLATE-NOT-SUPP	0C10	The specified template includes operand/ Data Elements which are not supported by the receiver.

Table A-2 Error Status Codes (Part 2)

----- Severity Status = INFO-NOT-SPEC (A200) -----		
<u>Name</u>	<u>Code</u>	<u>Description</u>
UID-MISSING	0010	The UID is not specified.
SESS-ID-MISSING	0020	There is no SESSION-ID specified.
NO-FORMAT-SPEC	0030	There is no Format specified.
PARAM-MISSING	1000	Other parameter(s) are missing.
----- Severity Status = ALREADY-KNOWN (A700) -----		
<u>Name</u>	<u>Code</u>	<u>Description</u>
UID-EXIST	0010	The Data Object announced in a READY/SPI-REQUEST already exists at the receiver.
PREV-SUBMITTED	0110	The Data Object is already IMS-known at this location.
----- Severity Status = RULE-VIOLATION (A800) -----		
<u>Name</u>	<u>Code</u>	<u>Description</u>
INVALID-SESS-ID	0010	There is no GROUPCOPY session opened for the specified SESSION-ID.
SESS-TIMEOUT	0020	A timeout occurred during a GROUPCOPY session.
SESS-ID-UNKNOWN	0030	The GROUPCOPY session was not started correctly.
STATE-VIOLATION	01xx	The initiator is not allowed to change this State because it would violate SPI-defined or private rules. The only SPI-defined value for "xx" is: 10 = The State of the Data Object cannot be changed from "ARCHIVED".
----- Severity Status = SUBSEQUENT-ERROR (A900) -----		
<u>Name</u>	<u>Code</u>	<u>Description</u>
SUBMIT-FAILED	0010	The receiver was unable to SUBMIT successfully.

Table A-2 Error Status Codes (Part 3, concluded)

----- Severity Status = REFUSED (AA00) -----		
<u>Name</u>	<u>Code</u>	<u>Description</u>
TRANSMISS-REJ	0010	The receiver refuses the transmission of a Data Object to the MOVE-DESTINATION.
PERM-STORAGE	0020	The Data Object is on permanent storage and is not forgotten/deleted.
LAST-COPY	0030	The copy of the Data Object is the last copy.
LOC-NOT-AVAIL	0100	The location is currently not available.
----- Severity Status = SYNTAX-ERROR (C300) -----		
<u>Name</u>	<u>Code</u>	<u>Description</u>
ILLEGAL-TEMPLATE	0010	The template does not fit the SPI-defined syntax.

INDEX

- A -
 "ACCEPT" 22*
 Acquisition (DE) 15-16*, 18
 ACR-NEMA Data Elements 18*
 Admitting Diagnosis (DE) 15-16
 Allergy Indication (DE) 15, 26*
 Application Service name 22, 31
 <ASCII Text String> 24
 Attending Physician (DE) 15

- B -
 Bits Stored (DE) 15

- C -
 CANCEL (ACR) 22
 capability name 22*
 CCI = Composite Capability
 Identification
 CHANGESTATE/SPI 11-12*, 28-30
 <CLASS> 22
 CLASS (DE) 19, 30
 Coarse Capabilities (DE) 21*, 23
 command:
 Data Element types 28-29*
 name format (SPI) 22*
 shadow group 17*, 19*, 20
 COMPLETION-STATE (DE) 19, 30
 Composite Capability
 Identification 22-23*
 compression 6*, 9, 27*
 Compression Algorithm (DE) 26*
 CONFIGURE/SPI 11-12*, 21*, 23,
 28-30
 COPY/SPI 11-12*, 28-30
 COUNT (DE) 19, 31
 Country Code 3*, 13*

- D -
 Data Element 2
 description 17*
 for command shadow group 19
 for DESC 15-16
 number actual 6-7
 number effective 6-7
 number ranges 17*
 SPI, special use of 18
 Types 17*
 Data Object 2, 5
 Description 13, 14*
 identification 13*

modification history 25*
 Subtype (DE) 8*, 15-16, 25
 Type classification 10*
 Type (DE) 8*, 10*, 15-16, 19,
 25, 31
 DATA-OBJECT-SIZE (DE) 19, 20*
 Data Set Subtype (DE) 15-16, 18
 Data Set Type (DE) 15-16
 DCC = Data Country Code
 DESC:
 = Data Object Description
 class 14*
 Descriptor (DE) 25
 elements 14*
 fixed 14*
 for Data Object Type Folder 16*
 for Data Set Type Graphics 16*
 for Data Set Type Image 15*
 for Data Set Type Text 16*
 free-formatted 14*
 DESUBMIT/SPI 11-12*, 28-30
 DIALOG (ACR) 22
 DO# format 14*

- E -
 ECHO (ACR) 22
 Element 24*
 element number actual 6*, 20*
 element number effective 6*, 20*
 Enumerated Value 4*
 escape mechanism 4*
 SPI 30-31*
 escape value 4*
 <Even Integer> 24

- F -
 FIND (ACR) 22
 "FINE" 22*
 Fine Capability (DE) 21*, 23
 Folder 9
 Group Elements 27*
 Type (DE) 27, 31
 FORGET-LAST-COPY (DE) 19, 30
 FORGET/SPI 11-12*, 20*, 28-30
 Forgotten Status (DE) 19-20
 Format (DE) 21*, 23*

- G -
 GCBEGIN/SPI 11-12*, 22, 28-30
 GCEND/SPI 11-12*, 22, 28-30
 GET (ACR) 22

Group 24*
 length 17
 number ranges 17*
 GROUPCOPY (HLF) session 28

- H -
 hexadecimal number convention 3*

- I -
 Identifier, Unique 13*
 identifying information elements
 25*
 IE = Imaging Equipment
 IE# format 13*
 Image:
 coding 6*
 (DE) 15-16, 18
 Management Service 14
 Presentation Elements 26*
 Imaging Equipment 13
 IMS = Image Management Service
 IMS-FLAG (DE) 19, 30
 Institutional Department (DE) 15,
 21*
 "ISSUE" 22*

- L -
 Location List (DE) 19, 20*
 logical address 18*

- M -
 manufacturer-specific Data
 Elements 5*
 MAX-RESPONSES (DE) 19
 message 5
 Modality (DE) 15
 Modality Type (DE) 13, 21*, 31
 Modified Image Date (DE) 15
 Modified Image Identification
 (DE) 15
 Modified Image Time (DE) 15
 Move Destination Service (DE) 31
 MOVE/SPI 11-12*, 22, 28-30

- N -
 NUM-OBJECTS (DE) 19, 21*

- O -
 <Odd Integer> 24
 Organ (DE) 15-16, 26*

- P -
 PACS# format 13*, 18*
 PACS-IE Ident 18*, 21*
 Patient:
 Age (DE) 15-16
 Birthdate (DE) 15-16
 Identification (DE) 15-16
 Information Elements 26*
 Maiden Name (DE) 15-16
 Name (DE) 15-16
 Sex (DE) 15-16
 Pixel Shadow Group Elements 27*
 Pregnancy (DE) 15, 26*, 31
 private extensions 8
 Procedure Description (DE) 15-16

- Q -
 Query:
 Elements (DE) 21*, 24*
 Match Return (DE) 21*, 24*, 31
 Operators (DE) 21*, 24*, 31
 Template (DE) 19, 21*
 QUERY/SPI 11-12*, 20*, 28-30
 template example 21*

- R -
 Radiologist (DE) 15-16
 READY/SPI 11-12*, 22, 28-30
 Referring Physician (DE) 15
 Report Date (DE) 16, 25*
 Report Time (DE) 16, 25*
 RESPONSES-FLAG (DE) 21*, 23, 31
 Resulting Diagnosis (DE) 15-16, 25
 RETURN-INFO-FLAG (DE) 19, 31

- S -
 SEND (ACR) 22
 SEND/SPI 11-12*, 22, 28-30
 Series (DE) 15-16, 18
 Service (DE) 19, 31
 SESSION-ID (DE) 19
 <Shadow_Owner_Code> 24*

SPI:

Enumerated Values 30-31*
Error Status 32*
extensions to ACR-NEMA 2
Recognition Code (DE) 4-5*, 8,
19, 24*, 30*
Release 1: 11-12, 30
Status Code (DE) 30
Status Codes 32-34*, 35-37
STATE (DE) 19, 20*
State:
"ARCHIVED" 20
"REPORTED" 20
<State_label> 20*
Station Identification (DE)
15-16, 21*
Status Code (DE) 19
Study:
Date (DE) 15-16
(DE) 15-16
Time (DE) 15-16
SUBMIT/SPI 11-12*, 28-30

- T -

<TOTAL-SIZE> 20*
Type, Data Object:
Compressed Image 9-12*
Folder 9-12*
Graphics 10-12*
Identifier 10-12*
Image 8, 10-12*
Null 8, 10-12*
Other 8*
Private Image 9
Text 8, 10-12*

- U -

UID = Identifier, Unique
Unique Identifier (DE) 13*, 25
user-specific Data Element 5*

- W -

Wait Time (DE) 19, 20*



STANDARD PRODUCT INTERCONNECT

for

COMPATIBILITY OF DIGITAL IMAGING

NOT TO BE DISCLOSED

SPI Document 5

Communication Interconnect

Document Version No. 1 Date: 1987-12-01
Document Edition No. 1 Date: 1987-12-01

This document is part of the SPI Release defined in the current "SPI Valid Documents List".

The document consists of 5 pages excluding this cover page.

Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved even in the event of the grant of a patent or the registration of a utility model or design.

CONTENTS

1.0	INTRODUCTION	2
1.1	OBJECTIVES	2
1.2	SCOPE	2
1.3	APPLICABLE DOCUMENTS	3
2.0	RESOLUTION OF AMBIGUITY IN THE ACR-NEMA STANDARD	4
2.1	ACR-NEMA PARAGRAPH 7.3.2.1	4
2.2	ACR-NEMA PARAGRAPH 8.3.2.1	4
2.3	ACR-NEMA PARAGRAPH 8.3.2.4	4
2.4	ACR-NEMA PARAGRAPH 8.3.4	4
2.5	ACR-NEMA PARAGRAPH 8.4.3	4
2.6	ACR-NEMA PARAGRAPH 8.4.4	4
2.7	ACR-NEMA PARAGRAPH 8.5.4	4
	INDEX	5

CHAPTER 1

I N T R O D U C T I O N

1.1 OBJECTIVES

Communications are an integral part of providing the functionality of a Picture Archiving and Communication System (PACS). The protocols and low-level interconnect standards formalize the basic data transfer between the constituent Imaging Equipments (IEs) of a PACS. The major objective of interconnecting disparate IEs, possibly over networks interfaced via Network Interface Equipment (NIE), possibly all from different vendors, requires strict adherence to well-established communications standards.

Toward this objective, the SPI has been structured to conform to the principles of the ISO/OSI Model and to conform to the ACR-NEMA Standard. Document 2 describes the concepts of how the SPI implements this conformance.

1.2 SCOPE

Document 5 considers the Standard Communications Interconnect (SCI) in the SPI that represents the "lower" levels of communication and how the SCI relates to the ACR-NEMA Standard.

In terms of the ISO/OSI Model, SCI encompasses roughly layers 1 through 5, namely, Physical, Data Link, Network, Transport, and Session Layer. Actually, some of these layers are functionally reduced for particular configurations. A minimum Session Layer function, establishing a connection, constitutes the upper limit of the SPI scope.

In terms of the ACR-NEMA Digital Imaging and Communication Standard, SCI contains the ACR-NEMA Hardware and Protocols specification (ACR-NEMA Section 7) and the Data Exchange Protocol (ACR-NEMA Section 8).

The interface between SCI and the higher SPI levels explicitly uses the Command Structure, Section 6 of the ACR-NEMA Standard. The service SCI offers to the next higher layer consists of establishing connections to specified target NIEs, transporting ACR-NEMA-structured Messages to the destination specified in Group 0 of the Message, and delivering the ACR-NEMA-defined Response where applicable.

Since the beginning of SPI definition activities, the original SPI goals and the ACR-NEMA standardization have converged to such a degree that this SCI has been reduced to a mere reference to the ACR-NEMA Standard. SCI is therefore virtually synonymous with Sections 7 and 8 of the ACR-NEMA Standard.

Some minor ambiguities in the ACR-NEMA Standard are listed and resolved in Chapter 2. These represent the only pertinent differences between SPI and the ACR-NEMA Standard in the communications area.

1.3 APPLICABLE DOCUMENTS

The ACR-NEMA Standard is defined in the following document:
"Digital Imaging and Communications Standard", ACR-NEMA Standards Publication No. 300-1985.

The ISO/OSI Model is defined in the following document:
"Information Processing Systems - Open Systems Interconnection - Basic Reference Model", International Organization for Standardization, ISO 7498, 1983.

CHAPTER 2

RESOLUTION OF AMBIGUITY
IN THE ACR-NEMA STANDARD

2.1 ACR-NEMA PARAGRAPH 7.3.2.1

The "implementation-specific fixed time" shall not be used within SPI; all SPI implementations shall make use of the "uniformly distributed random time".

2.2 ACR-NEMA PARAGRAPH 8.3.2.1

In the last sentence, for "by the peers across the interface", should be read "by that device".

Since an intervening network may transform channel numbers between peers, the only criterion for a device selecting a channel number is the device's own usage of channel numbers.

2.3 ACR-NEMA PARAGRAPH 8.3.2.4

In the last sentence, to avoid confusion substitute "...by the destination device of the Open Channel Request or Reset Channel Request".

2.4 ACR-NEMA PARAGRAPH 8.3.4

Referring to the second sub-paragraph, it is to be understood that a Reset Channel Request constitutes a premature termination of a message transfer the same way a Close Channel Request would. Therefore, it is not legal to change channel parameters during transfer of a message.

2.5 ACR-NEMA PARAGRAPH 8.4.3

There is an obvious misprint in this paragraph, specifying logical and physical destination address twice: Delete sentences five and six.

2.6 ACR-NEMA PARAGRAPH 8.4.4

This piece of Authorized Engineering Information appears rather confusing and therefore shall be disregarded.

2.7 ACR-NEMA PARAGRAPH 8.5.4

Within SPI, delaying the acknowledgements for Open, Reset or Close Channel Requests shall not exceed 15 seconds. Any SPI device is allowed to "time out" if such an acknowledgement is not received within 20 seconds.

INDEX

- A -

ACR-NEMA ambiguities 3-4*

- C -

channel number selection 4*
Close Channel Request 4
Communication Interconnect 2*

- O -

Open Channel Request 4
OSI Model 2

- R -

Reset Channel Request 4

- S -

SCI = Standard Communications
Interconnect
Standard Communications
Interconnect 2*



STANDARD PRODUCT INTERCONNECT

for

COMPATIBILITY OF DIGITAL IMAGING

NOT TO BE DISCLOSED

SPI Document 6

Off-Line Media and Data Formats

Document Version No. 1 Date: 1987-12-01
Document Edition No. 1 Date: 1987-12-01

This document is part of the SPI Release defined in the current "SPI Valid Documents List".

The document consists of 49 pages excluding this cover page.

Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved even in the event of the grant of a patent or the registration of a utility model or design.

CONTENTS

1.0	INTRODUCTION	2
1.1	OBJECTIVES	2
1.2	REQUIREMENTS	3
1.3	SCOPE	3
1.4	OTHER CONSIDERATIONS	4
2.0	SCOPE OF COMPATIBILITY	5
2.1	PHYSICAL MEDIUM	5
2.1.1	Physical Characteristics	5
2.1.2	Recording Characteristics	5
2.2	VOLUME ORGANIZATION	6
2.2.1	Organization of Data Objects on a Volume	6
2.2.2	Data Object Access Method	8
2.2.3	Directories	9
2.2.4	Index Area	10
2.2.5	Volume Physical Mapping	10
2.3	DATA OBJECT FORMAT	10
2.4	FURTHER CONSIDERATIONS	10
2.4.1	Medium Layout and Block Numbering	10
2.4.2	Defect Management	11
2.4.3	SPI and Write Once Read Many (WORM) Optical Disks	12
3.0	12" OPTICAL DISK	13
3.1	MEDIUM CHARACTERISTICS	13
3.1.1	Physical Characteristics	13
3.1.2	Recording Characteristics	13
3.2	VOLUME ORGANIZATION - OVERVIEW	14
3.3	INDEX AREA STRUCTURE	14
3.3.1	SPI Volume Label	16
3.3.2	User Volume Label 1	18
3.3.3	User Volume Label 2	23
3.3.4	Directory Reference Table	24
3.4	DIRECTORIES	27
3.4.1	Mandatory Directory Structure	27
3.4.2	Optional Directory Structure	30
3.5	DATA OBJECTS	31
3.6	VOLUME PHYSICAL MAPPING AND UPDATING	32
3.6.1	Mandatory Directory Mapping	32
3.6.2	Optional Directory Mapping	37
3.6.3	Data Object Storage	39
3.6.4	General Layout and Updating	39
3.6.5	Data Types	43
4.0	5 1/4" OPTICAL DISK (to be supplied at a later date)	44
APPENDIX	SYMBOL LIST	45
INDEX		46

CHAPTER 1

I N T R O D U C T I O N

1.1 OBJECTIVES

This document describes the compatibility conventions within the scope of the Standard Product Interconnect (SPI) for the interchange of Data Objects on off-line data storage media.

The data organization and structures documented here serve primarily for data exchange between different Imaging Equipment (IE). Archival capabilities are also included in the discussion. This document is not intended to define an overall PACS archive compatibility standard or to support an Image Management System (IMS). The access structures described should, however, be suitable for simple local archival purposes (e.g., at PACS-independent modalities). Procedures for transformation and duplication between archival and exchange media can be minimized by using these structures and data organization.

This document specifies the necessary and sufficient parameters and structures for the removable storage media discussed, which if used, will help achieve interchangeability between different Imaging Equipment. These parameters refer exclusively to Data Objects and media specifications.

1.2 REQUIREMENTS

SPI establishes the following design considerations as the basis for compatibility of off-line media and their data formats:

1. The volume organization and the corresponding data structures shall provide fast access to particular Data Objects, e.g., images, required by the user.
2. The access mechanism shall support direct access via a Primary Key to minimize resources such as access time, transfer time, cpu time, and memory space necessary for processing.
3. Optional access via Secondary Keys should be provided with emphasis on flexibility.
4. To ensure usability of a volume after a system failure, volume layout and design of access structures shall ensure that directory reconstruction is possible.
5. The interchange of volumes between IEs shall be independent of the respective host systems or operating systems used.
6. The Data Objects shall be structured according to the SPI definitions.
7. The syntax given by Document 1 is used to describe the data formats.
8. When other international standards (ISO, ECMA, etc.) are applicable, these shall also be used.

1.3 SCOPE

These storage media are considered in the following specifications:

- 12" Write Once Optical Disk.
- 5 1/4" Write Once Optical Disk.

1.4 OTHER CONSIDERATIONS

Double quotes ("...") are normally used in the SPI documents; single quotes ('...') are used for special purposes in syntax definitions and inside text that is already quoted.

SPI ignores upper/lower case differences in ASCII strings used in command and Data Element text. The convention in other text is to capitalize commands and specific terminology for emphasis. Terms are not capitalized in general use. Thus, for example, "Data Object" and "Data Element" are capitalized, while "object" and "element" are not. A special case, "data set", remains uncapitalized (except in tables and special lists), in conformance with ACR-NEMA usage.

Dates that are not part of SPI-defined syntax are written in ISO-format, e.g., 24 September 1987 becomes 1987-09-24.

In the text of this document, the convention is adopted that a hexadecimal number is designated by appending the letter "H" to the hexadecimal digit string.

Throughout this document, references to integer fields and lengths imply "unsigned integers".

The symbol "Ø" is sometimes used to represent zero instead of "0" where "0" might be misinterpreted.

A list of symbols that appear in tables, figures, and text can be found in the Appendix.

CHAPTER 2

SCOPE OF COMPATIBILITY

This chapter presents generic specifications which, when adapted for specific media, will ensure data interchangeability and satisfy the requirements described in 1.2. Actual specifications are defined in the corresponding chapters of this document for each of the specific media.

2.1 PHYSICAL MEDIUM

The physical medium is defined by its physical characteristics and recording characteristics. For example, industry standards exist to enable Imaging Equipment of different manufacturers to "write once" and "read many" to/from a standard optical disk physical medium. Other standards address different optical disk media technologies, magnetic disk media and magnetic tape media. The following subsections outline the primary characteristics whose standardization makes physical level data interchange possible between like media.

2.1.1 Physical Characteristics

- Type definition, e.g.:
 - o Characteristics of the active layer(s).
 - o Substrate specification.
 - o Geometrical dimensions.

2.1.2 Recording Characteristics

- Recording density (e.g., bpi, tpi).
- Recording method (e.g., MFM, NRZI).
- Error detection and correction method (e.g., CRC, ECC).
- Track preformatting.
- Track organization.
- Sector organization (for disk media).
- Block organization.
- Medium defect management.

2.2 VOLUME ORGANIZATION

This section describes the logical organization of Data Objects as they shall be stored on a volume to enable their retrieval using a specific access method and corresponding data structures. These logical structures are defined uniformly for all SPI storage media. The necessary index structures and the physical mapping of these logical structures are defined for specific storage media in other chapters of this document.

2.2.1 Organization of Data Objects on a Volume

Each Data Object (DO) shall occupy a contiguous recording extent on the storage medium, except in the case of physical replacement of physically unusable recording areas following the medium-specific rules for defect management.

A directory entry shall not refer to an incomplete Data Object. Hence, only complete Data Objects are accessible via the directory. Data Objects cannot extend across volumes; thus, each directory entry specifies a Data Object completely contained on that volume.

Identification of Data Objects

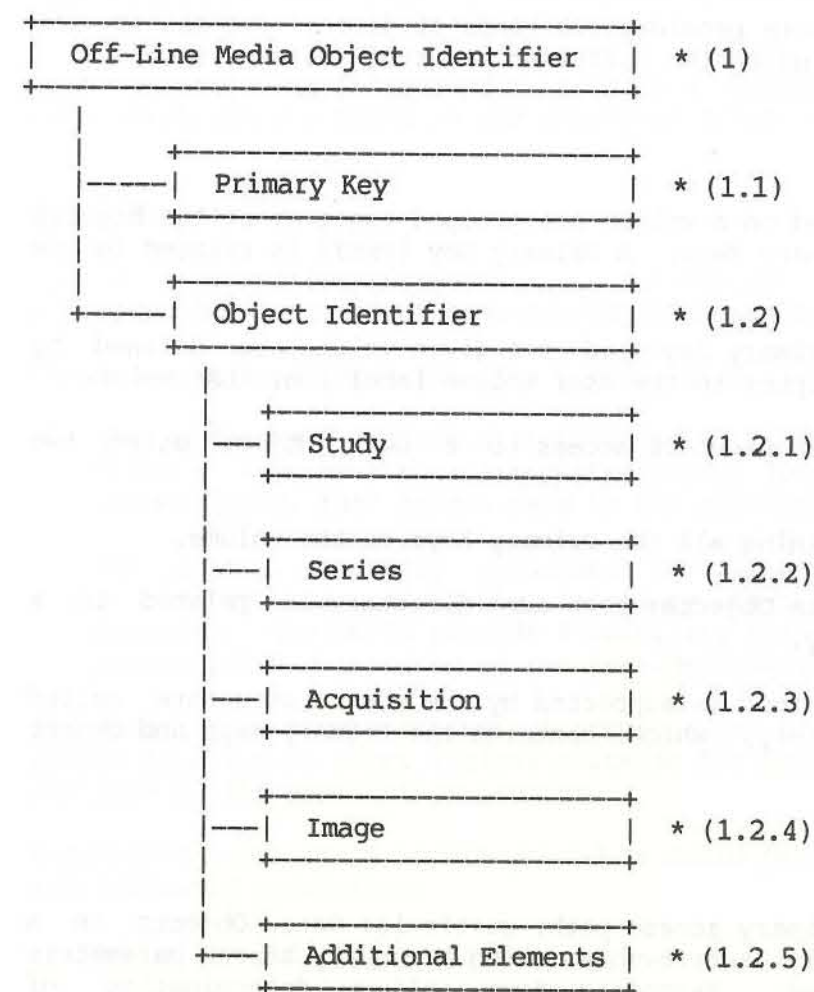
Each Data Object stored on a volume shall have an identifier which is unique within that volume. This identifier is used to retrieve the Data Object. It should be noted that the Data Object identification scheme formalized in this document is independent of any identification scheme required by an Image Management Service. Hence, this off-line media-oriented identifier is called the Off-Line Media Object Identifier (OMO_ID). As far as is appropriate, the elements of the OMO ID correspond with the elements of the object identification scheme of Documents 2, 3 and 4.

The OMO ID structure provides hierarchical ordering for sets or groups of Data Objects of four levels.

A defined subset of the OMO ID elements has a special importance to the access mechanism to particular Data Objects. This subset, called the Primary Key, is used as a primary search criterion for retrieving a Data Object from the volume (see 2.2.2).

An OMO ID value consists of a Primary Key value and an Object Identifier value. The OMO ID is composed of the following elements to ensure the required value uniqueness and to satisfy the access method described below:

Figure 2.2-1 Off-Line Media Object Identifier (OMO_ID)



These elements are determined by the following rules:

- * (1): Shall be unique within one particular volume.
- * (1.1): Identifies one Data Object or a series of Data Objects by higher order identification parameters (e.g., patient-oriented parameters). Implies a primary search criterion for Data Object(s) on a particular volume.
- * (1.2): Shall be unique within a particular Primary Key that is related to a particular Data Object.
- * (1.2.1-4): Each identifies one level of object order belonging to a particular Primary Key; this results in four hierarchy levels of object order per Primary Key.
- * (1.2.5): Can ensure uniqueness of OMO_ID and provide user-specific differentiation of Data Objects.

2.2.2 Data Object Access Method

The method described here provides two kinds of direct access to the desired Data Objects, using two different kinds of search criteria.

Primary Access

The Data Objects stored on a volume are grouped together at the highest level by means of Primary Keys. A Primary Key (PKEY) is related to one or more Data Objects.

The elements of the Primary Key used on a given volume are defined by the Primary Key Descriptor in the User Volume Label 1 of that volume.

The primary access path provides access to a Data Object using two structures:

- A directory containing all the Primary Keys on the volume.
- Directories of Data Objects; each such directory is related to a single Primary Key.

This primary access method is supported by the data structure called the Mandatory Directory, which contains the Primary Keys and Object Identifiers.

Secondary Access

In addition to the primary access path, particular Data Objects on a volume are optionally retrievable using secondary search parameters called Secondary Keys. Secondary Keys allow determination of application-specific selection criteria of Data Objects that are different from those of the Primary Key access.

A Secondary Key is represented by a Keyword which identifies a Data Element, and a related value (see Document 4). A Secondary Key is defined as:

```
<SKEY>      ::= <SKEY_WORD><SKVAL>
<SKEY_WORD> ::= <Group-nr><Element-nr>
```

The secondary access paths, each determined by a particular Secondary Key, shall be supported by one access data structure composed of corresponding relation lists. One complete set of such access data structures is called an Optional Directory.

Depending on the application, the user can decide which kinds of optional Secondary Keys and corresponding relations are to be written onto a volume.

This document does not intend to provide a general IMS-oriented data base functionality based on the secondary access mechanism described.

2.2.3 Directories

The two types of Data Object access mechanisms described above are supported by access structures stored on the volume.

The two main structures for this purpose are:

- The Mandatory Directory structure (MDIR), supporting access via the Primary Key and the associated Object IDs. The MDIR provides a standardized, fast access path to the selected Data Objects.
- The Optional Directory structure (ODIR), permitting alternative access by user-selectable Secondary Keys composed of SPI Data Elements. The ODIRs provide flexibility for supporting secondary access paths determined by the user-dependent application.

The updating procedure of the directory structures, with regard to update time and/or other logical criteria for updating, is unspecified, and left to the application program.

A particular directory update procedure could perform one or more of the following operations:

- Add a new Data Object to a volume:
 - o Whose Primary Key already appears in the directory.
 - o Whose Primary Key was not already in the directory.
- Add an ODIR Structure with defined Secondary Keys.
- Add new Secondary Keys to an already existing ODIR Structure.

2.2.4 Index Area

General information is stored in an Index Area in the first blocks of each volume for the purpose of volume identification, description, and directory references.

This Index Area contains four different data structures:

- SPI Volume Label - identifies a particular volume, and specifies general volume parameters.
- User Volume Label 1 - contains additional information regarding the directory structures defined by the user.
- User Volume Label 2 - contains volume status information after the final write operation on the volume has been performed (see 2.4.3).
- Directory Reference Table - provides an access path to the actual directories via pointers to their physical locations on the volume. When there is an Optional Directory, the Directory Reference Table also specifies those Secondary Key Words which are supported by the access structure, and contains pointers to the secondary key subdirectories.

2.2.5 Volume Physical Mapping

This section describes the layout and physical mapping of the optical disk volume. The main structures are:

- Index Area.
- Directories.
- Data Objects.

2.3 DATA OBJECT FORMAT

The Data Objects stored on an off-line medium shall conform to the definitions of Document 4 except no "UID" is used for off-line media. Note that Groups 0 and 1 are not part of a Data Object by definition.

2.4 FURTHER CONSIDERATIONS

2.4.1 Medium Layout and Block Numbering

A medium is organized physically by dividing surface area into blocks. A block is contiguous on a surface, of fixed size, and contains control and data areas whose detailed layout is specific to a given media standard. For the types of storage media considered in this document, blocks are numbered logically; the first logical block is block zero.

The relationship of logical to physical block numbers is defined by the detailed specifications for the medium drive and is not needed in this document. During a write/read operation, a drive writes/reads units of blocks of data to/from the medium at logical block locations on the medium.

2.4.2 Defect Management

Medium defect management (also known as bad block handling), is defined by the specifications for the medium/drive/controller employed.

For magnetic disks, the bad blocks are typically detected during an initial write/verify procedure, and their addresses are entered in a table written in a reserved area on the media surface. This table enables the controller to skip (and sometimes to remap) bad blocks on subsequent write or read accesses. This is not possible for Write Once Read Many (WORM) optical disks, because a write/verify process writes data, thus preventing future writing.

Bad blocks on a WORM disk are discovered while or after writing user data. Two methods of defect management exist at present: rewrite mode and reassign mode.

Rewrite Mode

When a bad block is detected during a write operation, it is marked as such, and then the data are rewritten to the following block. If this fails again, the drive tries to write the data to the next following block, and so on, for a limited number of times. This mode uses the Direct Read During Write (DRDW) feature that makes it possible to discover bad blocks while they are being written; no extra revolution of the disk is required for the verify operation.

Note that this method causes a difference between "logical" and "physical" disk block addresses for the replacement block and all following (higher-numbered) blocks. This difference increases with every new bad block processed. Note, further, that the occurrence of bad blocks reduces the disk space available to the user; this has to be taken into account when determining the extent of fixed-sized areas.

Reassign Mode

In this mode, spare blocks in reserved areas on the medium are used as substitutes for bad blocks. When a bad block is detected by the controller, it is marked as such, and the data are rewritten to a "spare" block. The address of the replacement is then entered in a special field in the bad block. In this mode, a difference between "logical" and "physical" block address applies only to the pairs of bad/replacement blocks.

2.4.3 SPI and Write Once Read Many (WORM) Optical Disks

Where WORM Optical Disks are considered, a brief description of terminology and SPI-compatible usage is helpful.

A WORM optical disk medium is initialized during manufacture with servo information along each track, and header information for each sector on a track. Typically each track has tens of sectors, and a medium has tens of thousands of tracks. A sector of a track contains the fixed size header, which contains control information, and a fixed sized data field, which is initially unwritten. This data field is the physical block, referred to above, and is sized to contain about 1000 bytes of user data. During write operations the physical blocks are written to, and the medium eventually can be filled up. As explained above, the successful writing of a logical block to a specific physical block maps that physical block to that logical block number for all further operations. The media defect management scheme guarantees that a logical block will be successfully written, and since read operations operate on previously written logical blocks, references to blocks below and in Chapter 3 refer to logical blocks.

SPI-compatible WORM media have volume labels near the physical beginning of each medium in reserved block locations. Labels are followed by some unwritten blocks, followed in turn by the Directory Reference Table. The Directory Reference Table has a reserved number of unwritten blocks set aside which get written as described below. Following the Directory Reference Table is a data area where Data Objects and directories are written with no empty blocks between or within Data Objects or directories.

The write-once characteristics of the WORM Optical Disk require writing new directories when new data are added. New data followed by new directories are written starting at the end of the old data area. The directories and Data Objects are ordered by design so that a directory shall follow the data it points to on the disk. Except for this restriction, Data Objects and directories are written to the data area on the volume in the order writes occur from the application. When writing in the data area is completed, the application writes information into the Directory Reference Table indicating the location of the now-current directory structure. This process of adding new data and changing and writing new relevant directories is referred to as "updating" the volume.

The design of an SPI-compatible application requires that careful consideration be given to how much data can be written in the Directory Reference Table and in the data area following it. An application shall not write beyond or "overrun" prescribed data areas. If a limit is reached, the WORM medium is "closed" by writing appropriate information in User Volume Label 2. A closed WORM volume can be used only for reading data.

CHAPTER 3

1 2 " OPTICAL DISK

3.1 MEDIUM CHARACTERISTICS

For the user's convenience, the next sections give a short summary of the main characteristics of the LaserDrive 1200 Optical Disk Medium. The LD 1200 media are characterized as Write Once Read Many or "WORM" media.

For detailed information, the user should refer to the following document and its referenced applicable documents: "LaserDrive 1200, Intelligent Digital Optical Disk Drive with SCSI, Product Specification", - Rev. February 1987.

3.1.1 Physical Characteristics

- Structure: Sandwich
- Laser sensitive layer: Tellurium alloy
- Substrate: Polymer on glass
- Surface(s): 1 or 2, pregrooved
- Track: Spiral

3.1.2 Recording Characteristics

- Recording method: Hole burning
- User accessible tracks: 32000
- Sectors/track: 32
- User bytes/sector: 1024
- Medium defect management: Scheme is Auto Rewrite

3.2 VOLUME ORGANIZATION - OVERVIEW

The following data structures are introduced to provide a suitable organization on the non-erasable LD 1200 disk:

- Index Area Structures.
- Directories.
- Data Objects.

A Data Object (DO) contains the relevant user data. Its format is defined in Document 4. Other structures serve for access and identification of the DOs and the identification and description of the recorded disk volume itself.

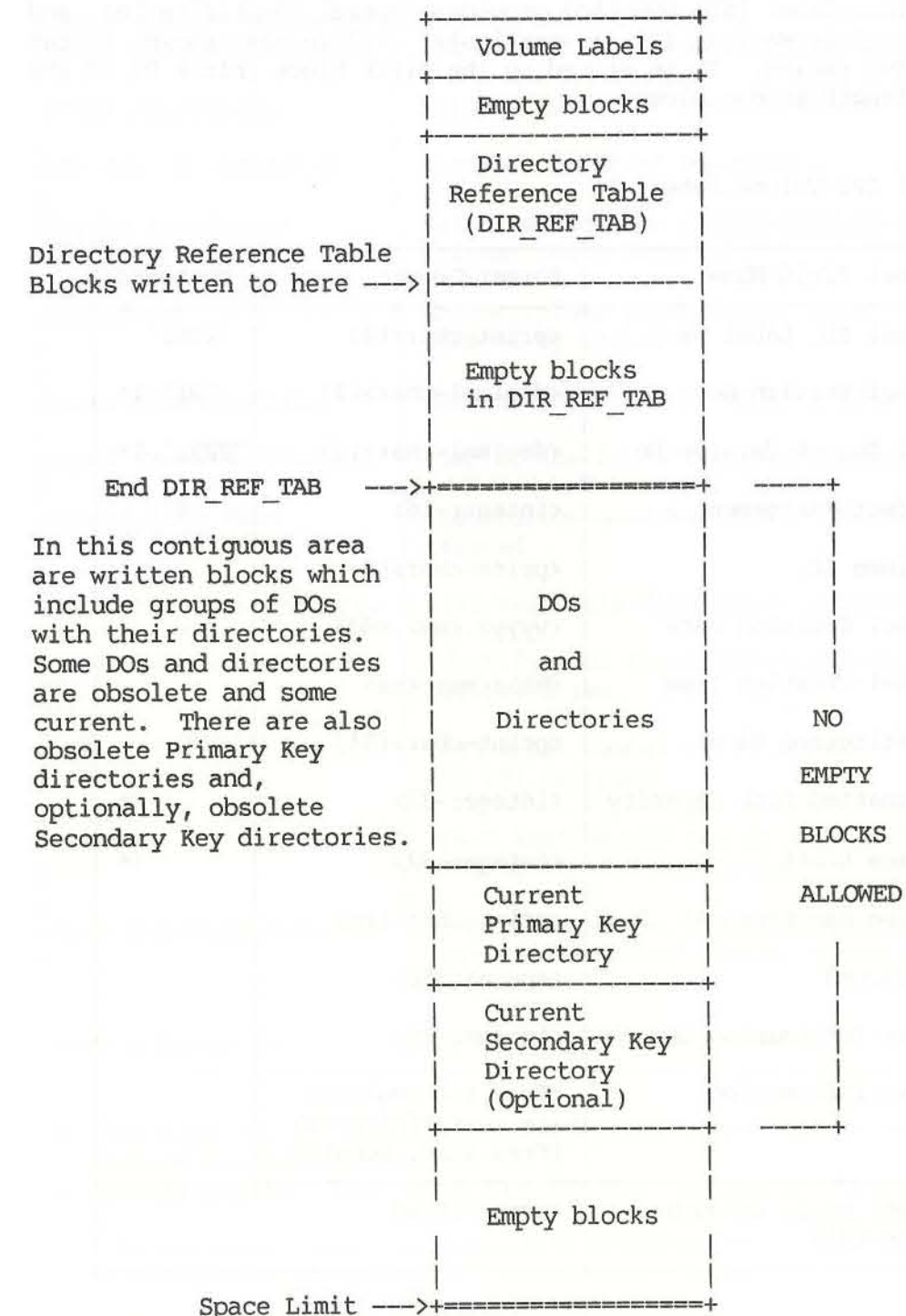
A side of an optical disk medium recorded with the above data structures in the way described in this document is called a volume. Thus, a double-sided disk would be two volumes.

A simplified layout of a volume is shown in Figure 3.2-1 (page 15).

3.3 INDEX AREA STRUCTURE

Three specific labels (the SPI Volume Label and User Volume Labels 1 and 2) and the Directory Reference Table form the Index Area located in the first blocks of the optical disk. The number of blocks reserved for the Directory Reference Table is defined by the user (see below, User Volume Label 1). The subsequent disk space, up to the Space Limit value written in the SPI Volume Label, may be used for user data, Data Objects, and the related directory structures.

Figure 3.2-1 Simplified Layout of a Disk Volume



3.3.1 SPI Volume Label

The SPI Volume Label (SPI_VOL_LAB) provides general identification and description information for a particular volume as shown in the following two tables. It is stored on the first block (block 0) of the disk; its length is one block.

Table 3.3-1 SPI Volume Label

Byte	Label Field Name	Format/Length	Contents
0	Label ID, Label No	<print-char>(4)	"VOL1"
4	Label Version No	<decimal-char>(2)	"01" 1*
6	SPI Doc. 6 Version No	<decimal-char>(4)	"0001" 1*
10	Defect Management	<integer-16>	0
12	Volume ID	<print-char>(8)	
20	Label Creation Date	<yyyy>.<mm>.<dd>	
30	Label Creation Time	<hh>:<mm>:<ss>	
38	Institution Name	<print-char>(64)	
102	Formatted Disk Capacity	<integer-32>	2*
106	Space Limit	<integer-32>	3*
110	Drive Manufacturer ID	<print-char>(16)	
126	Reserved	<space>(386)	
512	Free Information Length	<integer-16>	
514	Free Information	<Free Information> ::= <print-char> (Free Info. Length)	
	empty space through byte 1023	unspecified	

- 1* Different version numbers are required for Document 6 and the Label Version Number because the document may change while the label remains the same.
- 2* 1024000 blocks of 1024 bytes each for the LD 1200 in Auto Rewrite Mode.
- 3* Less than or equal to 1023999 for the LD 1200 in Auto Rewrite Mode.

Table 3.3-2 Meanings of the SPI Volume Label Entries

Label ID, Label No	label identifier and label number
Label Version No	version number of the SPI Volume Label
SPI Doc. 6 Version No	version number of Document 6
Defect Management	defect management method used on this volume: Auto Rewrite
Volume ID	unique volume identifier; to be created by the user
Label Creation Date	date of the label (volume) creation
Label Creation Time	time of the label (volume) creation
Institution Name	institution name where this volume was created
Formatted Disk Capacity	user-available disk space in blocks 1*
Drive Manufacturer ID	identifier of the manufacturer of the optical drive
Space Limit	last block number which may be used 2* for SPI purposes
Reserved	space that should not be used for user-optional information
Free Information Length	length, in bytes, of the user-optional "Free Information" field. Value shall be even.
Free Information	user-optional information

- 1* The capacity is reduced by bad blocks, which cannot be identified before the disk is written on.
- 2* Blocks beyond this number are for the implementor's private use.

IMPLEMENTATION NOTES

- Volume ID, Institution Name, Label Creation Date, and Label Creation Time together are intended to uniquely identify the particular volume.
- The implementor shall provide suitable protection features to prevent the Auto Rewrite mode of Defect Management from causing SPI data to overwrite the Space Limit.

3.3.2 User Volume Label 1

The User Volume Label 1 (USER_VOL_LABEL1) contains additional information to be defined by the user. This includes references to the Directory Reference Table and the formats of the descriptors used in the directory structure. User Volume Label 1 shall be written on the first block following the SPI Volume Label (block 1); its length is one block.

Table 3.3-3 User Volume Label 1

Byte	Label Field Name	Format/Length	Contents
0	Label ID, Label No	<print-char>(4)	"VOL2"
4	Label Version	<decimal-char>(2)	"01"
6	NOB Dir. Ref. Table	<integer-16>	
8	Dir. Ref. Tab. Pointer	<integer-16>	
10	Primary Key Descriptor	(36 bytes, see page 19)	
46	Object ID Descriptor	(90 bytes, see page 19)	
136	Reserved	<space>(376)	
512	Free Information Length	<integer-16>	
514	Free Information	<Free Information> ::= <print-char> (Free Info. Length)	
	empty space through byte 1023	unspecified	

Table 3.3-4 Meanings of the User Volume Label 1 Entries

Label ID, Label No	label identifier and label number
Label Version	version of the User Volume Label 1
NOB Dir. Ref. Tab.	number of blocks in the Directory Reference Table
Dir. Ref. Tab. Pointer	block address of the Directory Reference Table
Primary Key Descriptor	descriptor info. for Primary Keys used in the Primary Key Directory; see Table 3.3-5 below
Object ID Descriptor	descriptor information for the Object Identifiers used in the Object Directories; see Table 3.3-6 (page 21)
Reserved	space that should not be used for user-optional information
Free Information Length	length, in bytes, of the user-optional "Free Information" field
Free Information	user-optional information

The Primary Key (PKEY) Descriptor gives the user some flexibility in defining the Primary Key Values (PK_VALs) used as the access parameters in the Primary Key Directory (PK_DIR). For details see 3.6.1.

Table 3.3-5 PKEY Descriptor

Byte	Descriptor Field Name	Contents/Format	Description
0	PKEY_WORD Element 1	0010H,0010H <integer-16>(2)	ACR-NEMA code for Patient Name
4	PK_VAL Element 1 Length	<integer-16>	
6	PKEY_WORD Element 2	0010H,0030H <integer-16>(2)	ACR-NEMA code for Patient Birthdate
10	PK_VAL Element 2 Length	10 or 0 <integer-16>	

contd

Table 3.3-5 (contd) PKEY Descriptor

Byte	Descriptor Field Name	Contents/Format	Description
12	PKEY_WORD Element 3	0010H,0040H <integer-16>(2)	ACR-NEMA code for Patient Sex
16	PK_VAL Element 3 Length	2 or 0 <integer-16>	
18	PKEY_WORD Element 4	0010H,0020H <integer-16>(2)	ACR-NEMA code for Patient ID
22	PK_VAL Element 4 Length	<integer-16>	
26	PKEY_WORD Element 5	<integer-16>(2)	SPI code user-defined *
28	PK_VAL Element 5 Length	<integer-16>	
30	PKEY_WORD Element 6	<integer-16>(2)	SPI code user-defined *
34	PK_VAL Element 6 Length	<integer-16>	

* These elements shall be user-defined in accordance with Document 4 and the ACR-NEMA Standard. If the length field has the value zero, the value of the SPI code is irrelevant.

The PK_VAL Element Length shall be positive and may be zero. A zero length implies the element is omitted in the P_KEY. At least one PK_VAL element shall have a non-zero length.

****IMPLEMENTATION NOTE****

Two options are given for the purpose of referring to non-patient-oriented Data Objects:

1. Use of PKEY_WORD Element 4 with Pseudo-Patient ID.
2. Use of PKEY_WORD Elements 5 and 6 with any Data Object element code.

The OBJ_ID Descriptor allows the user to design the Object Identifiers (OBJ_IDs) which are used as access parameters in the Object Directories (OBJ_DIRS). For details see 3.6.1.

Table 3.3-6 OBJ_ID Descriptor

Byte	Descriptor Field Name	Contents/Format	Description
0	OBJ_ID Element 1 Name	0008H,1010H <integer-16>(2)	ACR-NEMA code for Station ID
4	OBJ_ID Element 1 Length	<integer-16>	
6	OBJ_ID Element 2 Name	0008H,0040H <integer-16>(2)	ACR-NEMA code for Data Object Type 1*
10	OBJ_ID Element 2 Length	<integer-16>	
12	OBJ_ID Element 3 Name	0020H,0010H <integer-16>(2)	ACR-NEMA code for Study
16	OBJ_ID Element 3 Length	<integer-16>	
18	OBJ_ID Element 4 Name	0008H,0020H <integer-16>(2)	ACR-NEMA code for Study Date
22	OBJ_ID Element 4 Length	10 or 0 <integer-16>	
24	OBJ_ID Element 5 Name	0008H,0030H <integer-16>(2)	ACR-NEMA code for Study Time
28	OBJ_ID Element 5 Length	8 or 0 <integer-16>	
30	OBJ_ID Element 6 Name	0020H,0011H <integer-16>(2)	ACR-NEMA code for Series
34	OBJ_ID Element 6 Length	<integer-16>	
36	OBJ_ID Element 7 Name	0020H,0012H <integer-16>(2)	ACR-NEMA code for Acquisition
40	OBJ_ID Element 7 Length	<integer-16>	
42	OBJ_ID Element 8 Name	0020H,0013H <integer-16>(2)	ACR-NEMA code for Image
46	OBJ_ID Element 8 Length	<integer-16>	
48	OBJ_ID Element 9 Name	0020H,3402H <integer-16>(2)	ACR-NEMA code for Modified Image ID
52	OBJ_ID Element 9 Length	<integer-16>	

contd

Table 3.3-6 (contd) OBJ_ID Descriptor

Byte	Descriptor Field Name	Contents/Format	Description
54	OBJ_ID Element 10 Name	0020H,3403H <integer-16>(2)	ACR-NEMA code for Modified Image Date
58	OBJ_ID Element 10 Length	10 or 0 <integer-16>	
60	OBJ_ID Element 11 Name	0020H,3405H <integer-16>(2)	ACR-NEMA code for Modified Image Time
64	OBJ_ID Element 11 Length	8 or 0 <integer-16>	
66	OBJ_ID Element 12 Name	<integer-16>(2)	SPI code user-defined 2*
70	OBJ_ID Element 12 Length	<integer-16>	
72	OBJ_ID Element 13 Name	<integer-16>(2)	SPI code user-defined 2*
76	OBJ_ID Element 13 Length	<integer-16>	
78	OBJ_ID Element 14 Name	<integer-16>(2)	SPI code user-defined 2*
82	OBJ_ID Element 14 Length	<integer-16>	
84	OBJ_ID Element 15 Name	<integer-16>(2)	SPI code user-defined 2*
88	OBJ_ID Element 15 Length	<integer-16>	

1* Data Object Type is synonymous with ACR-NEMA Data Set Type.

2* These elements shall be user-defined in accordance with Document 4 and the ACR-NEMA Standard. If the length field has the value zero, the value of the SPI code is irrelevant.

The OBJ_ID Element i Length shall be positive and may be zero. A zero length implies the element is omitted in the OBJ_ID. At least one OBJ_ID element shall have a non-zero length.

****IMPLEMENTATION NOTE****

The element lengths defined in the PKEY Descriptor and the OBJ_ID Descriptor restrict the corresponding directory entries to exactly those lengths.

3.3.3 User Volume Label 2

User Volume Label 2 (USER VOL LAB2) shall be recorded directly after User Volume Label 1 when the volume is closed. Once closed, the volume shall be disabled from further recording (in the SPI area). User Volume Label 2 shall contain the date of volume closing and a two-byte Volume Completion Status. This label also gives the user the option of storing private information. Its length is one block.

Table 3.3-7 User Volume Label 2

Byte	Label Field Name	Format/Length	Contents
0	Label ID, Label No	<print-char>(4)	"VOL3"
4	Label Version	<decimal-char>(2)	"01"
6	Volume Closing Date	<yyyy>.<mm>.<dd>	
16	Completion Status	<integer-16>	
18	Reserved	<space>(494)	
512	Free Information Length	<integer-16>	
514	Free Information	<Free Information> ::= <print-char> (Free Info. Length)	
	empty space through byte 1023	unspecified	

Table 3.3-8 Meanings of the User Volume Label 2 Entries

Label ID, Label No	label identifier and label number
Label Version	version of the User Volume Label 2
Volume Closing Date	date of the volume closing
Completion Status	volume completion code; see page 24
Reserved	space that should not be used for user-optional information
Free Information Length	length, in bytes, of the user-optional "Free Information" field. Value shall be even.
Free Information	user-optional information

Completion 0000H - Volume properly closed.
 Status 0010H - Directory Reference Table overflow.
 Codes 0012H - ODIR not, or incompletely, updated.
 0013H - PKEY DIR not, or incompletely, updated.
 0014H - OBJ_DIR not, or incompletely, updated.

3.3.4 Directory Reference Table

The Directory Reference Table (DIR REF TAB) provides direct access to the directory structures (see 3.4) which may be spread over the whole disk surface, and to the start of the user-available unwritten area.

As shown in Figure 3.3-1 (page 25), the Directory Reference Table is sequentially recorded starting from the block referenced by the Directory Reference Table Pointer in User Volume Label 1.

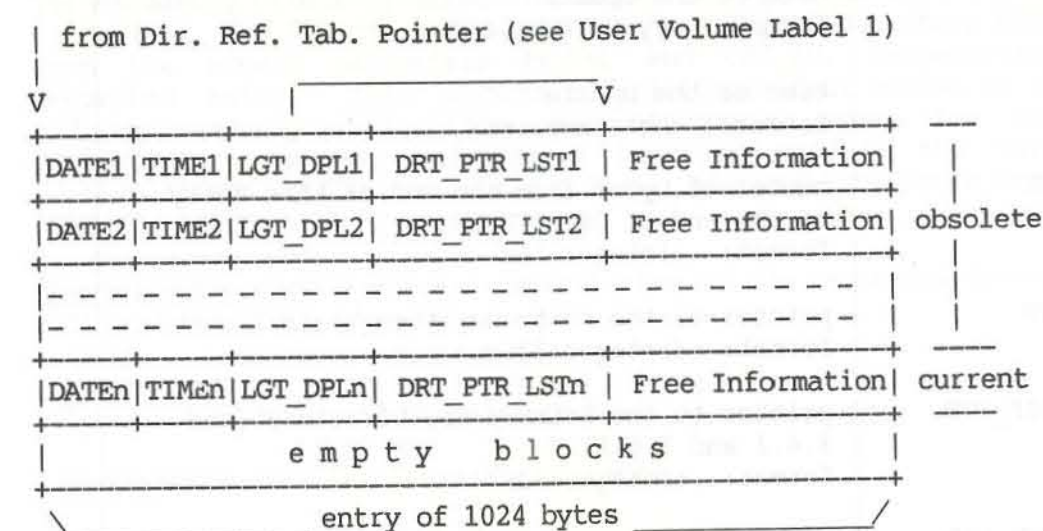
The Directory Reference Table has a user-defined fixed size which shall be written into the User Volume Label 1 at volume creation. Each entry of the Directory Reference Table is one block; the first entry starts in block zero of the Directory Reference Table. At each volume update, a new entry is added after the previously written entry. Every entry within the Directory Reference Table shall point to a (possibly obsolete) Primary Key Directory (PK_DIR). However, the last entry shall point to the current PK_DIR.

IMPLEMENTATION NOTES

1. The Directory Reference Table Size should be carefully determined (and entered as NOB Dir. Ref. Tab. into User Volume Label 1) in order to avoid an overrun during a volume update before the volume is closed. An adequate number of disk blocks should be reserved to accommodate media defect management.
2. Applications must search for the last entry of the Directory Reference Table to access currently-valid data on the volume.
3. In the case of a system crash, however, the obsolete pointers in the earlier entries may be useful for volume restoration.

The Directory Reference Table has the following structure:

Figure 3.3-1 Directory Reference Table Structure



The Directory Reference Table Pointer List (DRT_PTR_LST) is composed as follows:

Figure 3.3-2 Directory Reference Table Pointer List

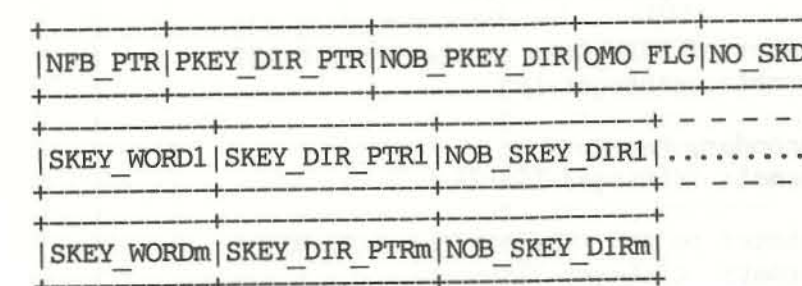


Table 3.3-9 Meanings and Formats of the Directory Reference Table Entries

DATE	date of the update format: <yyyy>.<mm>.<dd>
TIME	time of the update format: <hh>:<mm>:<ss>
LGT_DPL	number of bytes from the end of this entry to the end of DRT_PTR_LST format: <integer-16>
NFB_PTR	pointer to the next unwritten block format: <integer-32>
PKEY_DIR_PTR	pointer to the Primary Key Directory (see 3.4.1 and 3.6.1) format: <integer-32>
NOB_PKEY_DIR	number of blocks in the Primary Key Directory format: <integer-16>
OMO_FLG	Off-Line Media Object Identifier Flag that indicates whether PK_VAL and OBJ_ID are present in the Optional Directory (see 3.6.2) contents: "YY" if present; "NN" if omitted format: <print-char>(2)
NO_SKD	number of SKEY_DIRS in the DRT_PTR_LST format: <integer-16>
SKEY_WORD	Secondary Key Word *
SKEY_DIR_PTR	pointer to the SKEY DIR for SKEY_WORD format: <integer-32>
NOB_SKEY_DIR	number of blocks in the above-mentioned SKEY DIR format: <integer-32>
Free Information	user-optional information format: unspecified

* The value of SKEY_WORD here shall be the same as SKEY_WORD in the SKEY_DIR to which SKEY_DIR_PTR points. See 3.4.2 and 3.6.2 for Optional Directory structure and mapping, respectively.

IMPLEMENTATION NOTE

The block size of 1024 bytes constrains the number of SKEYs.

NOT TO BE DISCLOSED

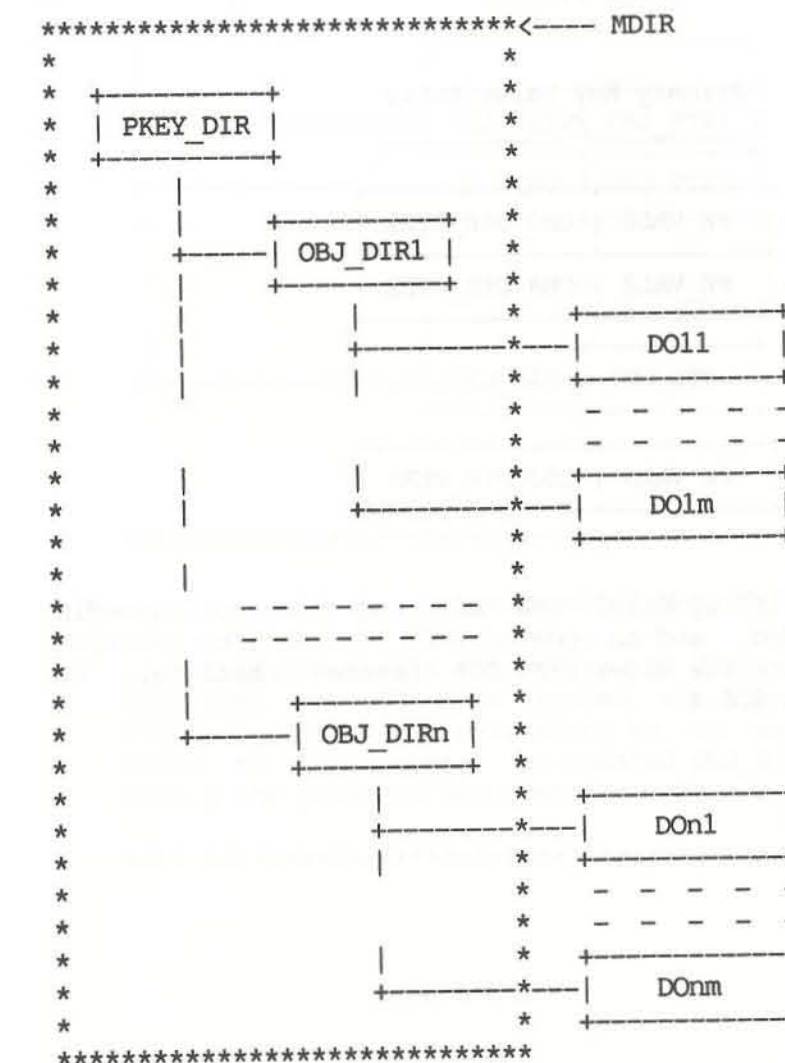
3.4 DIRECTORIES

3.4.1 Mandatory Directory Structure

The Mandatory Directory Structure (MDIR) is organized into two levels, a Primary Key Directory (PKEY_DIR) and an Object Directory (OBJ_DIR), with the access parameters PK_VAL and OBJ_ID, respectively, as described below. This structure facilitates fast access to the DOs, and requires only small portions of host memory space for directory processing. PK_VAL and OBJ_ID together provide the unique DO identification on the particular volume: the Off-Line Media Object ID (OMO_ID) (see 2.2.2).

The following figure gives a general view of the relations between the MDIR and the DOs.

Figure 3.4-1 MDIR and DO Relationships



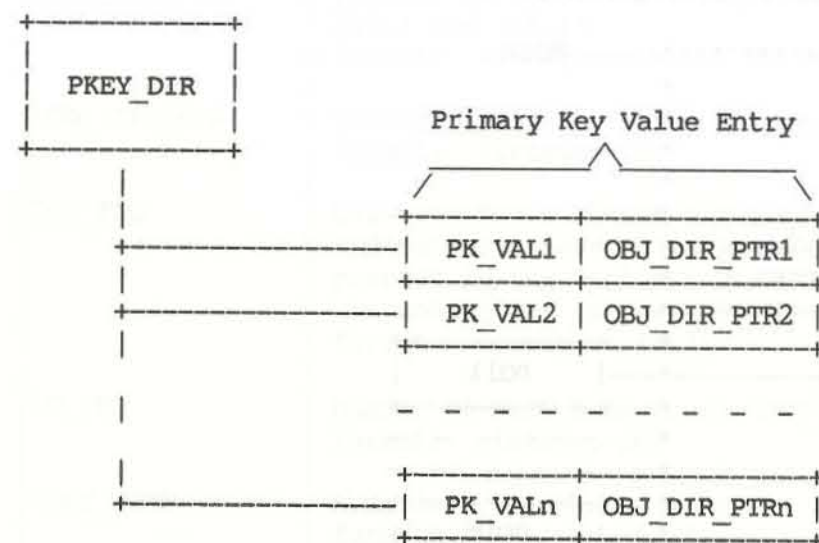
NOT TO BE DISCLOSED

Primary Key Directory (PKEY_DIR)

The PKEY_DIR is directly referenced by the PKEY_DIR_PTR in the Directory Reference Table (see 3.3.4) and links the PK_VALs with the Object Directories (OBJ_DIRS) by means of Object Directory Pointers (OBJ_DIR_PTRs). Each PK_VAL contains the essential identification information. No PK_VAL shall occur more than once in the PKEY_DIR. For details of the internal PK_VAL structure see Table 3.3-5 (page 19) and 3.6.1. There shall be exactly one current PKEY_DIR on the volume (see 3.6.4), through which all OBJ_DIRS are accessible.

The logical structure of the PKEY_DIR is shown in the following figure.

Figure 3.4-2 PKEY_DIR Logical Structure



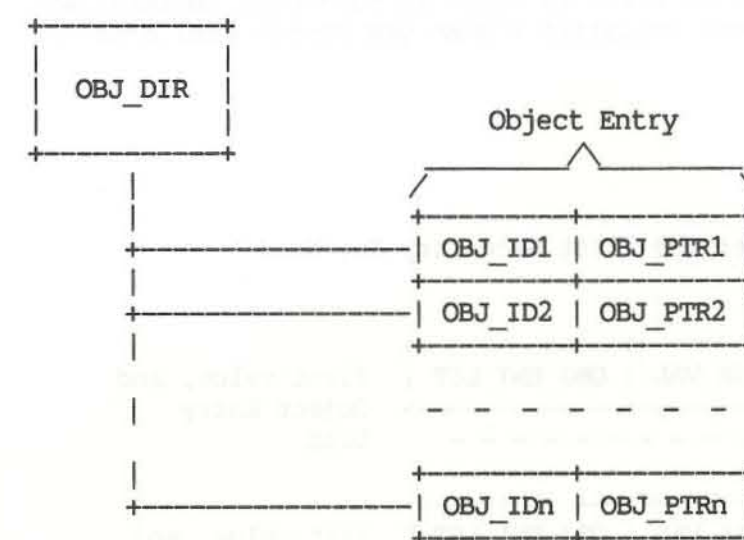
Every time the volume is updated, new DOs and the corresponding OBJ_DIRS shall be added, and an updated PKEY_DIR shall be recorded. After this PKEY_DIR update the older PKEY_DIR becomes obsolete. For further details refer to 3.6.4.

Object Directory (OBJ_DIR)

The OBJ_DIRS are directly referenced by the OBJ_DIR_PTRs, and link the OBJ_IDs with the DOs by means of Object Pointers (OBJ_PTRs). Thus, each OBJ_DIR corresponds to a particular PK_VAL. Under a particular PK_VAL, no OBJ_ID shall occur more than once. For OBJ_ID structure see Table 3.3-6 (page 21) and 3.6.1.

The logical structure of the OBJ_DIR is shown in the following figure.

Figure 3.4-3 OBJ_DIR Logical Structure



****IMPLEMENTATION NOTE****

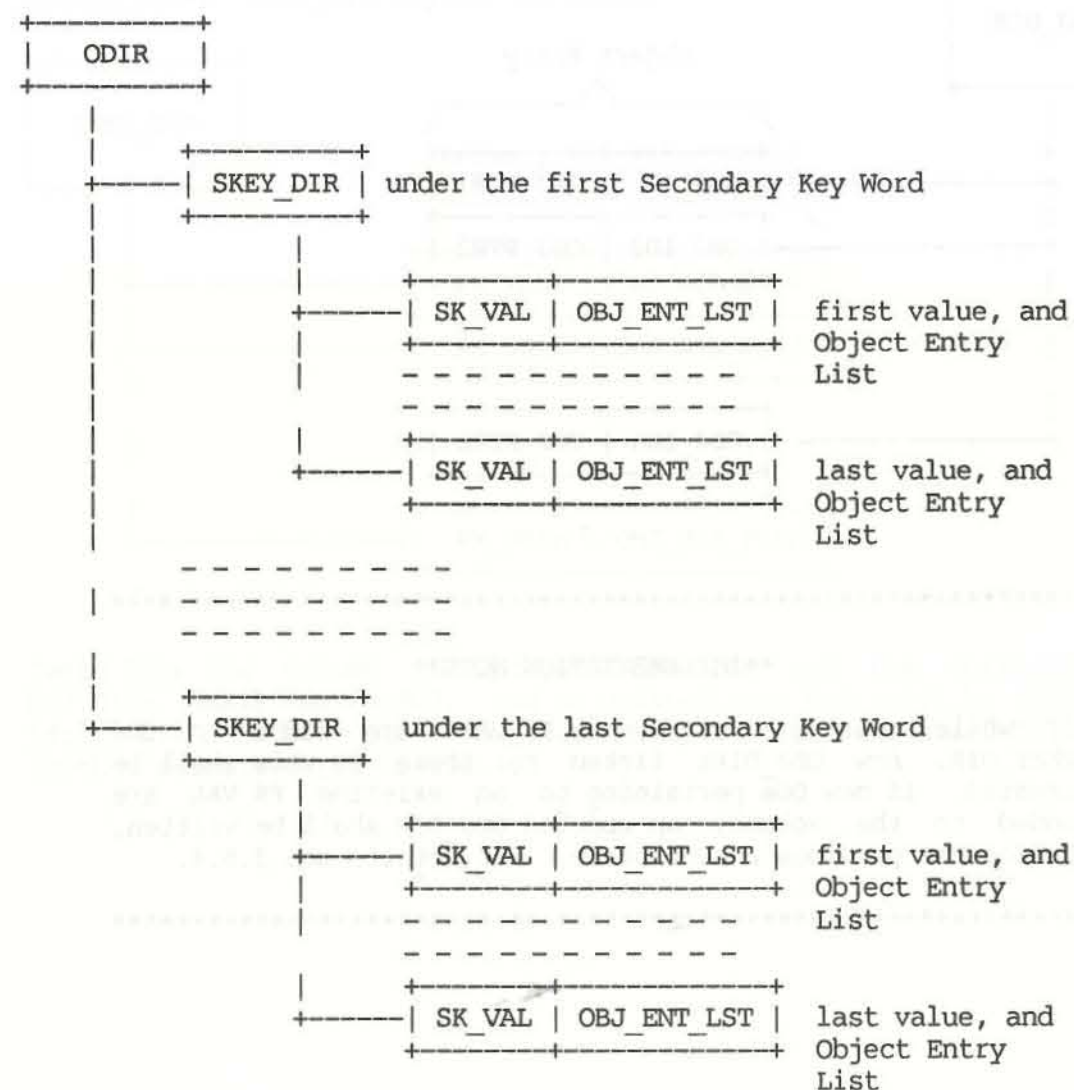
If, while updating a volume, new PK_VALs are added to the PKEY_DIR, new OBJ_DIRS linked to these PK_VALs shall be created. If new DOs pertaining to an existing PK_VAL are added to the volume, an updated OBJ_DIR shall be written, making the previous one obsolete. For details see 3.6.4.

3.4.2 Optional Directory Structure

The Optional Directory Structure (ODIR) enables the user to create optional relations which provide access via additional classification or selection criteria to Data Objects which satisfy these criteria. These criteria are called Secondary Keys. Each Secondary Key shall be composed of a Secondary Key Word, and an associated Secondary Key Value, which form a SKEY WORD/SK VAL pair.

A general view of the ODIR is given in the following figure.

Figure 3.4-4 ODIR Layout



Two alternative forms of the ODIR are provided. In the simpler one, only the OBJ_PTRs are listed after the Secondary Key Values (SK_VALs); for the more complex form, the OBJ_ENT_LST includes the OMO_ID and the OBJ_PTRs. The user shall indicate a choice by setting the Off-Line Media Object Identifier Flag (OMO_FLG) in the Directory Reference Table. Details of ODIR mapping are given in 3.6.2.

3.5 DATA OBJECTS

User data shall be recorded only in the form of Data Objects (DOs). The internal structure of these is based on the definitions in Document 4. Each Data Object may have a different length.

3.6 VOLUME PHYSICAL MAPPING AND UPDATING

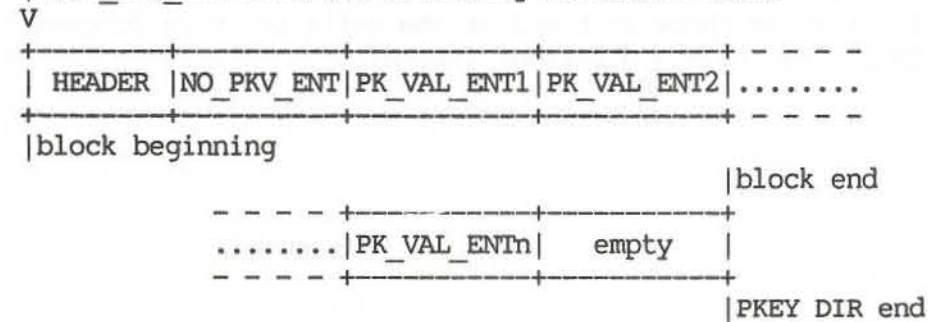
3.6.1 Mandatory Directory Mapping

Primary Key Directory (PKEY DIR)

The PKEY_DIR shall be contiguous, starting from a block boundary, as follows:

Figure 3.6-1 PKEY DIR Data Sequence

|PKEY DIR PTR from the Directory Reference Table



The Primary Key Value Entry (PK VAL ENT) has the following form:

Figure 3.6-2 Primary Key Value Entry Format

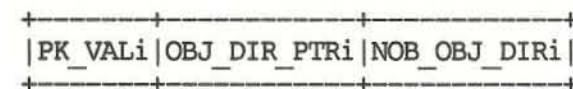


Table 3.6-1 Meanings and Formats of the Entries of the PKEY DIR

HEADER	PKEY_DIR Header format: <HEADER> ::= <TYPE><NOB><Reserved>
TYPE	directory type format: <print-char>(4) contents: PKEY
NOB	number of blocks occupied by the PKEY_DIR format: <integer-32>
Reserved	contents: 0 format: <integer-32>
NO_PKV_ENT	number of Primary Key Value Entries format: <integer-16>
PK_VAL_ENT	Primary Key Value Entry format: see Figure 3.6-2 (above)

contd

Table 3.6-1 (contd) Meanings and Formats of the Entries of the PKEY_DIR

PK_VAL	value under the PKEY_WORD, patient identifier information; structured in accordance with the PKEY Descriptor in User Volume Label 2; see Table 3.6-2 below. format: $\langle \text{PK_VAL} \rangle ::= \langle \text{PK_VAL Element 1} \rangle \langle \text{PK_VAL Element 2} \rangle \langle \text{PK_VAL Element 3} \rangle \langle \text{PK_VAL Element 4} \rangle \langle \text{PK_VAL Element 5} \rangle \langle \text{PK_VAL Element 6} \rangle$
PK_VAL Element	PK_VAL Element format: see Table 3.6-2 below
OBJ_DIR_PTR	Object Directory Pointer format: <integer-32>
NOB_OBJ_DIR	number of blocks occupied by the Object Directory format: <integer-32>

The elements of the PKEY WORD are defined in accordance with the corresponding ACR-NEMA Elements (see Document 4).

Table 3.6-2 PKEY WORD Elements

PK_VAL Elements	PKEY_WORD Element 1*	PK_VAL Format 2*
PK_VAL Element 1	Patient Name	<print-char>(udl)
PK_VAL Element 2	Patient Birthdate	<yyyy>.<mm>.<dd>
PK_VAL Element 3	Patient Sex	<print-char>(2); contents: "M", "F", or "O", followed by a blank
PK_VAL Element 4	Patient ID	<print-char>(udl)
PK_VAL Element 5	SPI user-defined	As defined in Document 4 for the elements being used
PK_VAL Element 6	SPI user-defined	

1* The PKEY_WORD Elements are entered as ACR-NEMA codes in the PKEY Descriptor which forms a part of the User Volume Label 1.

2* "udl" means user-defined length (see User Volume Label 1, Primary Key Descriptor, PK VAL Element 1 to 6 Lengths).

****IMPLEMENTATION NOTES****

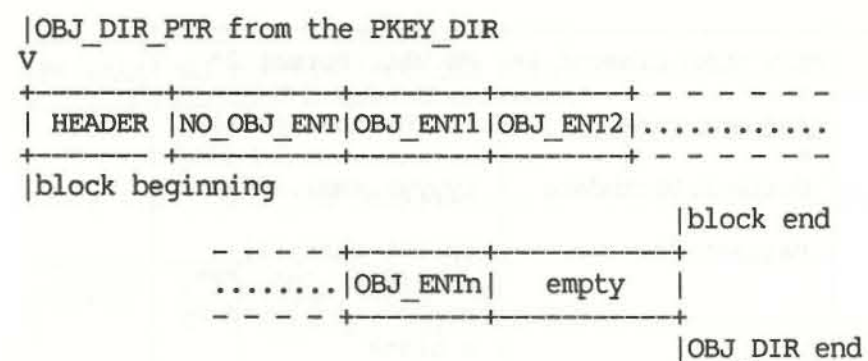
1. Elements of the PKEY_WORD can be eliminated by setting their lengths to zero in User Volume Label 1 at volume creation (see 3.3.2).
2. The values for PK_VAL Element 3 ("M ", "F ", and "O ") have the meanings "male", "female", and "other", respectively, as defined in the ACR-NEMA Standard. The abbreviation characters are followed by a blank to maintain an even byte count.

A PK_VAL is considered a character string resulting from a concatenation of the PK_VAL elements listed above in the order listed. The PK_VALs within the PKEY_DIR shall appear in ascending order according to the ASCII collating sequence.

Object Directory (OBJ_DIR)

The OBJ_DIR shall be contiguous, starting from the beginning of a block, as follows:

Figure 3.6-3 OBJ_DIR Contents



The Object Entry (OBJ_ENT) has the following form:

Figure 3.6-4 OBJ_ENT Contents

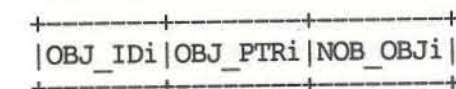


Table 3.6-3 Meanings and Formats of the Entries of the OBJ_DIR

HEADER	OBJ_DIR Header format: <HEADER> ::= <TYPE><NOB><Reserved>
TYPE	directory type format: <print-char>(4); contents: "DOBJ"
NOB	number of blocks occupied by the OBJ_DIR format: <integer-32>
Reserved	format: <integer-32>; contents: 0
NO_OBJ_ENT	number of DO entries in the OBJ_DIR format: <integer-16>
OBJ_ID	Data Object Identifier, structured in accord with the Object ID Descriptor in User Volume Label 1; see Tale 3.3-6 (page 21) <OBJ_ID> ::= <OBJ_ID Element 1><OBJ_ID Element 2>... ...<OBJ_ID Element 15>
OBJ_ID Element	OBJ_ID Element format: see Table 3.6-4 (page 36)
OBJ_PTR	Data Object Pointer; address of the first block of the DO format: <integer-32>
OBJ_ENT	Data Object Entry format: see Figure 3.6-4 above
NOB_OBJ	number of blocks occupied by the DO referenced format: <integer-32>

The structure of the OBJ_ID is defined by the OBJ_ID Descriptor in the User Volume Label 1 (see 3.3.2).

Table 3.6-4 OBJ_ID Elements

OBJ_ID Elements	OBJ_ID Element Name	Format 1*, 2*
OBJ_ID Element 1	Station ID	<print-char>(udl)
OBJ_ID Element 2	Data Object Type	<integer-16>
OBJ_ID Element 3	Study	<print-char>(udl)
OBJ_ID Element 4	Study Date	<yyyy>.<mm>.<dd>
OBJ_ID Element 5	Study Time	<hh>:<mm>:<ss>
OBJ_ID Element 6	Series	<print-char>(udl)
OBJ_ID Element 7	Acquisition	<print-char>(udl)
OBJ_ID Element 8	Image	<print-char>(udl)
OBJ_ID Element 9	Modif. Image ID	<print-char>(udl)
OBJ_ID Element 10	Modif. Image Date	<yyyy>.<mm>.<dd>
OBJ_ID Element 11	Modif. Image Time	<hh>:<mm>:<ss>
OBJ_ID Element 12	User-defined	As defined in
OBJ_ID Element 13	User-defined	Document 4 for
OBJ_ID Element 14	User-defined	the elements
OBJ_ID Element 15	User-defined	being used

- 1* The OBJ_ID Element Names are entered as ACR-NEMA codes in the OBJ_ID Descriptor which forms a part of the User Volume Label 1.
- 2* "udl" means user-defined length (see User Volume Label 1, Object ID Descriptor, OBJ_ID Element 1 to 15 Lengths).

IMPLEMENTATION NOTES

1. An element can be eliminated by setting its length to zero in the User Volume Label 1 at volume creation (see 3.3.2).
2. The definitions for the first 11 OBJ_ID Elements are selected from the ACR-NEMA defined Element Names (Group/Element) in accordance with Document 4.

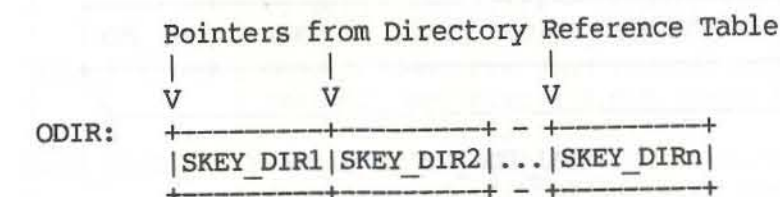
The OBJ_ID Elements 12 through 15 are user-definable in accordance with Document 4. They can be used to store additional DO specific information in the OBJ_DIR which can then be accessed without accessing the DO itself.

An OBJ_ID is considered a character string resulting from a concatenation of the OBJ_ID elements listed above in the order listed. The OBJ_IDs within the OBJ_DIR shall appear in ascending order according to ASCII collating sequence.

3.6.2 Optional Directory Mapping

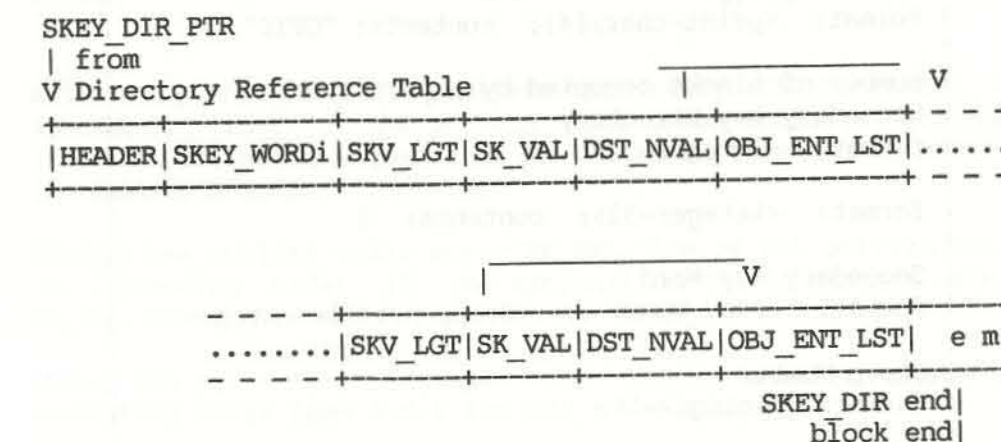
The ODIR is formed by a sequence of Secondary Key Directories (SKEY DIRs) created under a SKEY_WORD as shown below. For each SKEY DIR, the Directory Reference Table contains the corresponding Secondary Key Directory Pointer (SKEY_DIR_PTR) (see 3.3.4).

Figure 3.6-5 ODIR Contents



The SKEY DIRs shall be contiguous, starting from the beginning of a block, as follows:

Figure 3.6-6 SKEY_DIR Contents



Depending on the OMO_FLG setting ("NN" or "YY"), the Object Entry List does or does not include the OMO_ID as shown in the following figures:

Figure 3.6-7 OBJ_ENT_LST without OMO_ID, i.e., OMO_FLG Set to "NN":

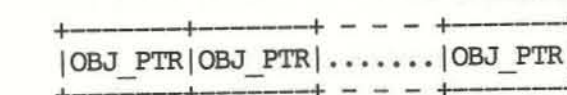


Figure 3.6-8 OBJ_ENT_LST with OMO_ID, i.e. OMO_FLG Set to "YY":

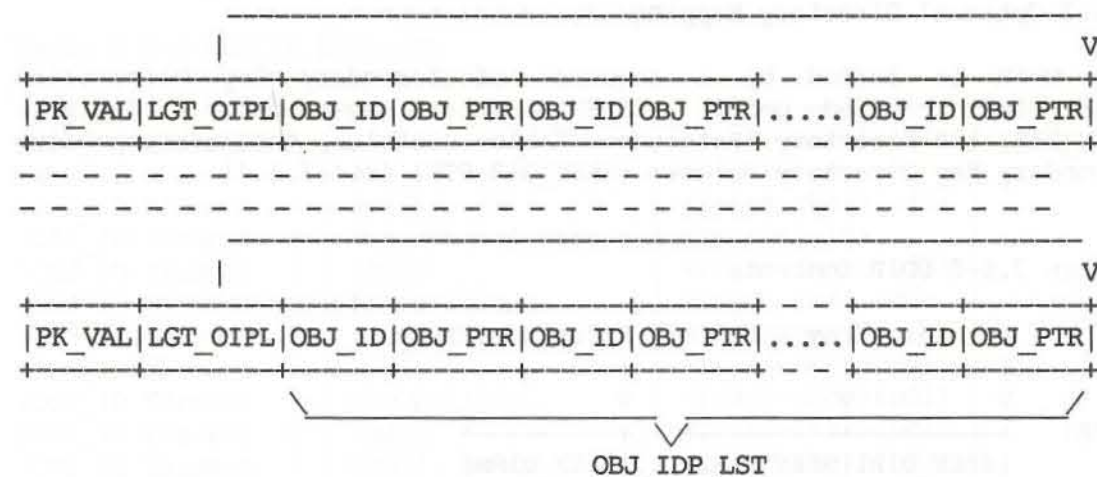


Table 3.6-5 Meanings and Formats of the Entries of the SKEY_DIR

HEADER	SKEY_DIR Header format: <HEADER> ::= <TYPE><NOB><Reserved>
TYPE	directory type Format: <print-char>(4); contents: "OPTI"
NOB	number of blocks occupied by a particular Secondary Key Directory format: <integer-32>
Reserved	format: <integer-32>; contents: 0
SKEY_WORD	Secondary Key Word format: <SKEY_WORD> ::= <Group-nr><Element-nr>
GROUP NO	Group Number format: <integer-16>
ELEMENT NO	Element Number format: <integer-16>
SKV_LGT	SK_VAL Length format: <integer-16>
SK_VAL	value under a Secondary Key word format: according to Document 4
DST_NVAL	distance to the next VAL, i.e. number of bytes from the end of this entry to the next SK_VAL format: <integer-32>

contd

Table 3.6-5 (contd) Meanings and Formats of the Entries of the SKEY_DIR

OBJ_ENT_LST	Object Entry List for a particular PK_VAL format: see Figures 3.6-7 (page 37) and 3.6-8 (page 38)
PK_VAL	value under the Primary Key Word; usually patient identification information format: see 3.6.1 - Primary Key Directory
OBJ_IDP_LST	Object Identifier and Pointer List format: see Figure 3.6-8 (page 38)
LGT_OIPL	length of the OBJ_IDP_LST, i.e. number of bytes from the end of this entry to the next PK_VAL format: <integer-16>
OBJ_ID	Object Identifier format: see Table 3.6-3 (page 35)
OBJ_PTR	Object Pointer format: <integer-32>; see Table 3.6-3 (page 35)

Within any SKEY_DIR, the sublists for every SK_VAL shall appear in ascending order of the SK_VALs. For these sublists, sorting shall be in ASCII collating sequence if SK_VAL is represented as ASCII text, and in numeric sequence otherwise.

Within the sublist under every SK_VAL, the PK_VAL entries shall appear in ascending order if the optional PK_VAL is present (see 3.6.1 for PK_VAL sorting).

Under the particular PK_VALs, the OBJ_ID entries shall appear in ascending order (see 3.6.1 for OBJ_ID sorting).

3.6.3 Data Object Storage

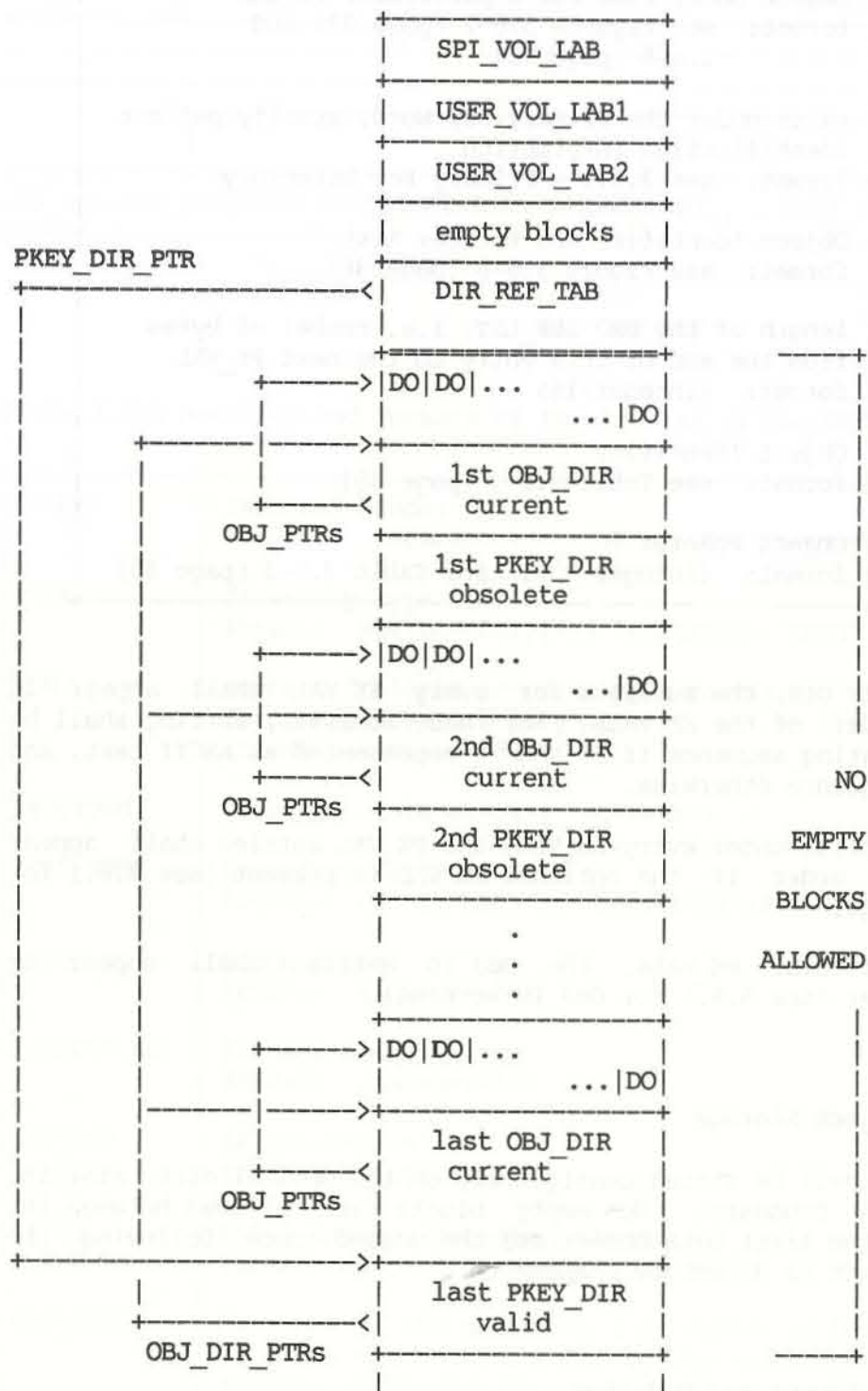
Data Objects shall be stored contiguously on the optical disk starting from a block boundary. No empty blocks are allowed between the beginning of the first Data Object and the unused space following the last Data Object or directory component.

3.6.4 General Layout and Updating

This section shows the physical layouts of the optical disk volume and explains them in the form of a short summary of the detailed description given in Chapter 3.

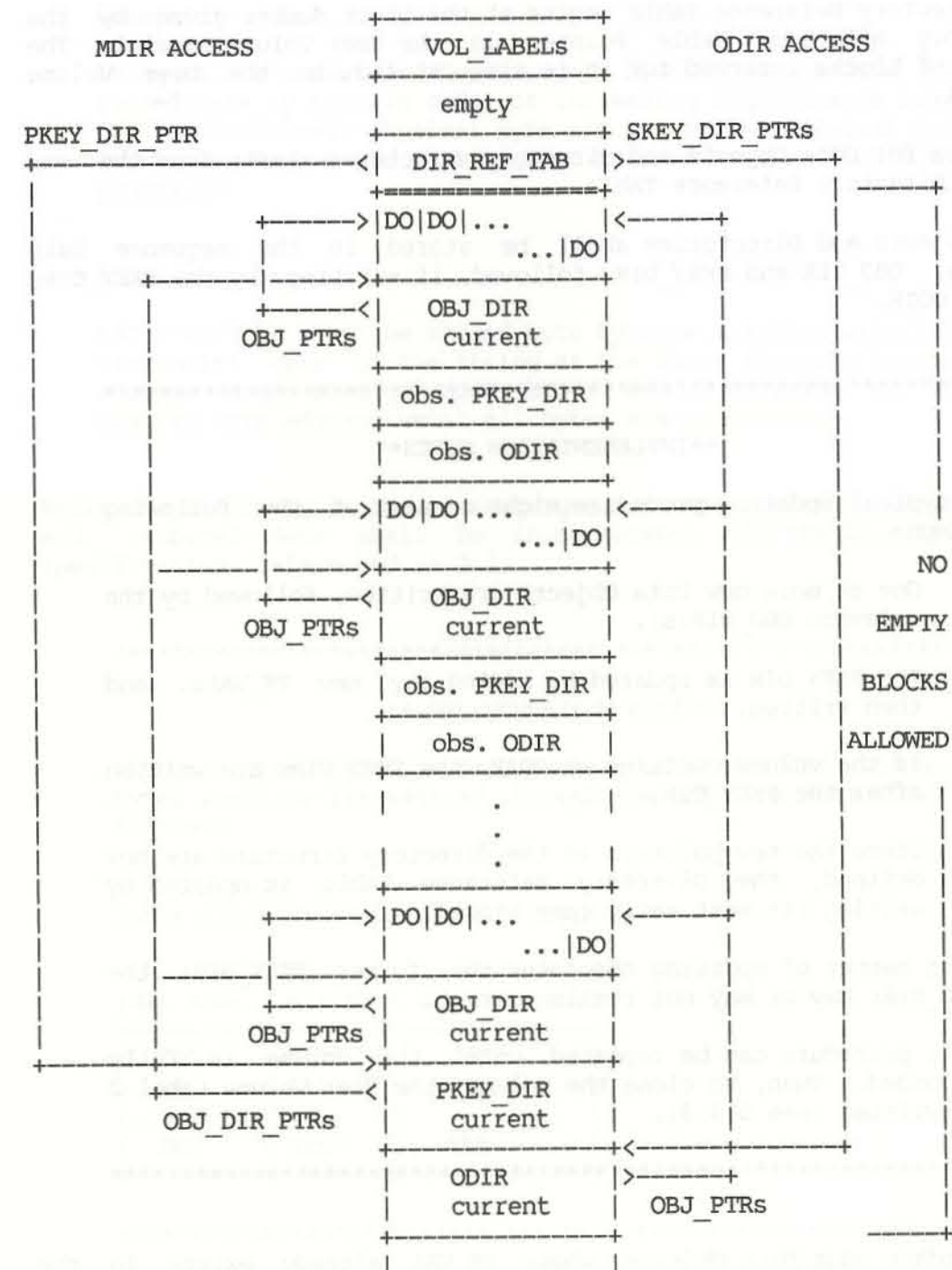
The following figure gives the layout of the physical organization of the optical disk volume without ODIR.

Figure 3.6-9 Disk Layout without ODIR



If an ODIR is created, it is written behind the MDIR, and the volume shows the following mapping:

Figure 3.6-10 Disk Layout with ODIR



The first three blocks of a volume shall be occupied by the SPI Volume Label, User Volume Label 1, and (when the volume has been closed) User Volume Label 2.

The SPI Volume Label and User Volume Label 1 shall be recorded when the volume is initialized by the application. For details see 3.3.1 and 3.3.2.

The Directory Reference Table begins at the block number given by the Directory Reference Table Pointer in the User Volume Label 1. The number of blocks reserved for it is also stored in the User Volume Label 1.

The area for Data Objects and directory structures starts from the end of the Directory Reference Table.

Data Objects and Directories shall be stored in the sequence Data Objects, OBJ_DIR and PKEY_DIR, followed, if existing, by the SKEY_DIRS of the ODIR.

IMPLEMENTATION NOTES

A typical updating procedure might consist of the following steps:

1. One or more new Data Objects are written, followed by the relevant OBJ_DIR(s).
2. The PKEY_DIR is updated by adding any new PK_VALs, and then written.
3. If the volume contains an ODIR, the SKEY_DIRS are written after the PKEY_DIR.
4. Since the new pointers to the directory structure are now defined, the Directory Reference Table is updated by writing its next entry (one block).

This manner of updating obsoletes the former PKEY_DIR; the OBJ_DIRS may or may not remain current.

This procedure can be repeated until the volume is fully recorded. Then, to close the volume, the User Volume Label 2 is written (see 3.3.3).

If an update adds Data Objects whose PK_VAL already exists in the directory, the old OBJ_DIR becomes obsolete. Entries corresponding to the old Data Objects shall be made in the new OBJ_DIR. Thus, the new OBJ_DIR contains references to the old, as well as the new Data Objects accessible via that PK_VAL.

3.6.5 Data Types

Data values shall be mapped on the optical disk medium as follows:

- Integer:

Integers are unsigned. Constituent bytes of integers shall be stored byte by byte in order of increasing significance starting at the first (lowest) physical byte address on the physical disk block and continuing at the next highest byte address until all bytes are processed.

- ASCII strings:

ASCII strings shall be stored byte by byte starting with the first (leftmost) byte of the string at the first (lowest) physical byte address on the physical disk block and continuing at the next highest byte address until all bytes are processed.

Media compatibility is the ultimate goal of this document. Toward this end, recorded data shall be in accordance with the LaserDrive 1200 specifications referenced in 3.1.

IMPLEMENTATION NOTE

If byte transmission to/from media uses the SCSI standard, bytes shall be transmitted through the SCSI Bus data lines as follows:

Table 3.6-6 Bit Transmission Mapping

Bit Name	Bit No	Significance
DB0	Bit0	LSB
DB1	Bit1	
...	...	
DB7	Bit7	MSB

CHAPTER 4

5 1/4" OPTICAL DISK

To be supplied in a later SPI Release.

APPENDIX

SYMBOL LIST

The following abbreviations are used in this document:

<u>SYMBOL</u>	<u>DEFINITION</u>
DIR_REF_TAB	Directory Reference Table
DO	Data Object
DRT_PTR_LST	Directory Reference Table Pointer List
DST_NVAL	Distance to Next Secondary Key Value
LGT_DPL	Length of the Directory Pointer List
LGT_OIPL	Length of Object ID and Pointer List
MDIR	Mandatory Directory structure
NFB_PTR	Next Free Block Pointer
NOB_OBJ	Number of Blocks of the Object
NOB_OBJ_DIR	Number of Object Directory blocks
NOB_PKEY_DIR	Number of Primary Key Directory blocks
NOB_SKEY_DIR	Number of Secondary Key Directory blocks
NO_OBJ_ENT	Number of Object Directory Entries
NO_PKV_ENT	Number of Primary Key Value Entries
NO_SKD	Number of Secondary Key Directories
OBJ_DIR	Object Directory
OBJ_DIR_PTR	Object Directory Pointer
OBJ_ENT	Object Entry
OBJ_ENT_LST	Object Entry List
OBJ_ID	Object Identifier
OBJ_IDP_LST	Object ID and Pointer List
OBJ_PTR	Object Pointer
ODIR	Optional Directory structure
OMO_FLG	Off-Line Media Object Identifier Flag
OMO_ID	Off-Line Media Object Identifier
PKEY	Primary Key
PKEY_DIR	Primary Key Directory
PKEY_DIR_PTR	Primary Key Directory Pointer
PKEY_WORD	Primary Key Word
PK_VAL	Primary Key Value
PK_VAL_ENT	Primary Key Value Entry
SKEY	Secondary Key
SKEY_DIR	Secondary Key Directory
SKEY_DIR_PTR	Secondary Key Directory Pointer
SKEY_WORD	Secondary Key Word
SKV_LGT	Length of a Secondary Key Value
SK_VAL	Value for a Secondary Key Word
SPI_VOL_LAB	SPI Volume Label
USER_VOL_LAB1	User Volume Label 1
USER_VOL_LAB2	User Volume Label 2

Note: - PTRs are pointers (absolute logical block numbers) to optical disk blocks. Byte distances are not necessarily counted from block boundaries; for them, the abbreviations LGT and DST are used. Thus, PTRs count in blocks, LGTs and DSTs in bytes.

I N D E X

- A -
 Acquisition (DE) 7*, 21, 36
 Auto Rewrite Mode 16

- B -
 bad block handling = defect management
 bit transmission mapping 43*
 block:
 logical 11*
 physical 11*
 byte transmission 43*

- D -
 Data Object 6, 45
 access method 8*
 Entry 35*
 format (optical disk) 10, 31
 identification 6
 Identifier (OMO_ID) 6-7, 35
 Pointer 35*
 Type (DE) 21-22, 36
 DATE 25*, 26
 Defect Management 16*, 17
 defect management 6, 11*, 24
 Direct Read During Write 11
 Directory Reference Table 10*, 12, 18, 24-26*, 28, 31-32, 37, 45
 Pointer 18*, 19, 24*, 25
 Pointer List 25-26*, 45
 Size 24
 directory (optical disk):
 reconstruction 3
 structure 27*
 update procedures 9
 DIR_REF_TAB = Directory Reference Table
 disk layout:
 and updating 39-41*
 with Optional Directory Structure 41*
 without Optional Directory Structure 40*
 Distance to Next Secondary Key Value 37, 38*, 45
 DO = Data Object
 DRDW = Direct Read During Write
 Drive Manufacturer ID 16*, 17
 DRT_PTR_LST = Directory Reference Table Pointer List

DST_NVAL = Distance to Next Secondary Key Value

- E -
 ELEMENT NO 38
 <Element-nr> 8, 38

- F -
 Formatted Disk Capacity 16*, 17
 Free Information 16*, 17, 18*, 19, 23*, 26
 Length 16*, 17, 18*, 19, 23*

- G -
 GROUP NO 38
 <Group-nr> 8, 38

- H -
 HEADER 32*, 34, 35*, 37, 38*

- I -
 Identifier, Off-Line (OMO_ID) = Off-Line Media Object Identifier
 Image (DE) 7, 21, 36
 Index Area 10*
 Institution Name 16*, 17

- K -
 Key, Primary 3, 6-9*, 19*, 27, 45
 Key, Secondary 3, 8*, 9-10, 30*, 45

- L -
 Label:
 Creation Date 16*, 17
 Creation Time 16*, 17
 ID 16*, 17, 18*, 19, 23*
 No 16*, 17, 18*, 19, 23*
 Version 18*, 19, 23*
 Version No 16*, 17
 LD 1200 Laser Drive 13*, 16
 Length of a Secondary Key Value 37, 38*, 45
 Length of Object Identifier and Pointer List 38, 39*, 45

Length of the Directory Pointer List 25-26*, 45
 LGT_DPL = Length of the Directory Pointer List
 LGT_OIPL = Length of Object Identifier and Pointer List

- M -
 Mandatory Directory 8*
 mapping 32*
 Structure 9, 27*, 45
 MDIR = Mandatory Directory Structure
 media:
 layout and block numbering 10-11*
 physical characteristics 5*, 13*
 recording characteristics 5*, 13*
 Modified Image:
 Date (DE) 22, 36
 Identification (DE) 21, 36
 Time (DE) 22, 36

- N -
 Next Free Block Pointer 25-26*, 45
 NFB_PTR = Next Free Block Pointer
 NOB = Number of blocks
 NOB_OBJ = Number of Blocks of the Object
 NOB_OBJ_DIR = Number of Object Directory blocks
 NOB_PKEY_DIR = Number of Primary Key Directory blocks
 NOB_SKEY_DIR = Number of Secondary Key Directory blocks
 NO_OBJ_ENT = Number of Object Directory Entries
 NO_PKV_ENT = Number of Primary Key Value Entries
 NO_SKD = Number of Secondary Key Directories
 Number of blocks 32*, 35*, 38*, 45
 Number of Blocks:
 of Directory Reference Table 18*, 19, 24
 of the Object 35*, 45
 Number of Object Directory:
 blocks 32-33*, 45
 Entries 34, 35*, 45

Number of Primary Key:
 Directory blocks 25-26*, 45
 Value Entries 32*, 45
 Number of Secondary Key:
 Directories 25-26*, 45
 Directory blocks 25-26*, 45

- O -
 OBJ_DIR = Object Directory
 OBJ_DIR_PTR = Object Directory Pointer
 Object Directory 27-28, 29*, 34*, 45
 contents 34*
 Entry meanings and formats 35*
 Pointer 28-29, 32-33*, 34, 45
 Pointer logical structure 29*
 Object Entry 34, 35*, 45
 List 30-31, 37-38, 39*, 45
 Object Identifier 21*, 35*, 38, 39*, 45
 and Pointer List 38, 39*, 45
 Descriptor 18*, 19, 21-22*, 35-36
 Element 35-36*
 Element "#n" 36*
 Element "#n" Length 21-22*
 Element "#n" Name 21-22*
 ordering 36
 Object Pointer 29, 31, 35*, 37-38, 39*, 45
 OBJ_ENT = Object Entry
 OBJ_ENT_LST = Object Entry List
 OBJ_ID = Object Identifier
 OBJ_IDP_LST = Object Identifier and Pointer List
 OBJ_PTR = Object Pointer
 ODIR = Optional Directory Structure
 off-line media 2*
 compatibility requirements 3*
 data representations 43*
 Off-Line Media Object Identifier 6-7*, 27, 31*, 37, 45
 Flag 25-26*, 31, 37*, 45
 structure 6
 OMO_FLG = Off-Line Media Object Identifier Flag
 OMO_ID = Off-Line Media Object Identifier
 optical disk 3, 10, 13*

Optional Directory 8*, 10, 37*
 alternative forms 31*
 contents 37*
 layout 30*
 mapping 37*
 Structure 9, 30*, 45

- P -

Patient:
 Birthdate (DE) 19, 33*
 Identification (DE) 20, 33*
 Name (DE) 19, 33*
 Sex (DE) 20, 33*
 permanent storage 2
 PKEY = Key, Primary
 PKEY_DIR = Primary Key Directory
 PKEY_DIR_PTR = Primary Key
 Directory Pointer
 PKEY_WORD = Primary Key Word
 PK_VAL = Primary Key Value
 PK_VAL_ENT = Primary Key Value
 Entry
 primary access method 8*
 Primary Key Descriptor 8*,
 18-19*, 19, 33
 Primary Key Directory 19-20*, 24,
 27, 28*, 32*, 45
 data sequence 32*
 Entry meanings and formats
 32-33*
 logical structure 28*
 Pointer 25-26*, 28, 32, 45
 Primary Key Value 19*, 28,
 32-33*, 38, 39*, 45
 Element 33*
 Entry 32*, 45
 ordering 34
 Primary Key Word 33*, 45
 Element 33*

- R -

reassign mode (optical disk) 11*
 Reserved entry 32, 35, 38
 Reserved space 16*, 17-19, 23
 rewrite mode (optical disk) 11*

- S -

SCSI 13, 43
 secondary access method 8*

Secondary Key Directory 37*, 38,
 45
 contents 37*
 Entry meanings and formats
 38-39*
 Pointer 25-26*, 37, 45
 Secondary Key Value 30*, 31
 Secondary Key Word 8*, 25*, 26,
 30*, 37, 38*, 45
 Series (DE) 7, 21, 36
 SKEY = Key, Secondary
 SKEY_DIR = Secondary Key Directory
 SKEY_DIR_PTR = Secondary Key
 Directory Pointer
 SKEY_WORD = Secondary Key Word
 SK_VAL = Value for a Secondary
 Key Word
 SKV_LGT = Length of a Secondary
 Key Value
 Space Limit 14, 16*, 17
 SPI:
 Document 6 Version No 16*, 17
 Volume Label 10*, 45
 Volume Label entry meanings 17*
 SPI VOL LAB = SPI Volume Label
 Station Identification (DE) 21, 36
 Study:
 Date (DE) 21, 36
 (DE) 7, 21, 36
 Time (DE) 21, 36

- T -

TIME 25*, 26
 TYPE, directory 32*, 35*, 38*

- U -

User Volume Label 1: 8, 10*, 18*,
 24, 33, 35-36, 45
 entry meanings 19*
 User Volume Label 2: 10*, 23*, 45
 entry meanings 23*
 USER VOL LAB1 = User Volume
 Label 1
 USER VOL LAB2 = User Volume
 Label 2

- V -

Value for a Secondary Key Word
 37, 38*, 39, 45

volume:
 closing 12*
 layout 15*, 40*, 41-42
 organization 6*, 14-15*
 overrun 12, 24
 physical mapping 10*
 restoration after crash 24
 updating 12*, 28, 29*, 32*, 42*
 Volume:
 Closing Date 23*
 Completion Status 23-24*
 ID 16*, 17
 Label, SPI 16*

- W -

WORM = Write Once Read Many
 Write Once Read Many 3, 11*



STANDARD PRODUCT INTERCONNECT

for

COMPATIBILITY OF DIGITAL IMAGING

NOT TO BE DISCLOSED

Global Index

Document Version No. 1 Date: 1987-12-01
Document Edition No. 1 Date: 1987-12-01

This document is part of the SPI Release defined in the current "SPI Valid Documents List".

The document consists of 8 pages excluding this cover page.

Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved even in the event of the grant of a patent or the registration of a utility model or design.

- A -

"ACCEPT" 4:22*
 Acquisition (DE) 4:15-16*, 18, 6:7*, 21, 36
 ACR 1:2*
 ACR-NEMA:
 ambiguities 5:3-4*
 commands 3:56
 Data Elements 4:18*
 Standard 1:2*
 Admitting Diagnosis (DE) 4:15-16
 <A-int> 1:13*
 Allergy Indication (DE) 4:15, 26*
 <alpha-char> 1:12*
 <Alt> 1:14*
 <A-number> 1:13*
 Application:
 Command 1:2*, 2:11*, 17, 3:3, 16, 28, 36*, 37, 38*, 57*, 58
 Service 2:15*, 21, 3:5, 10*, 35-37, 57-58*
 Service name 3:10, 30, 4:22, 31
 <A-real> 1:13*
 <ascii-char> 1:12*
 <ASCII Numeric String> 3:24*
 <ASCII Text String> 3:24*, 4:24
 Attending Physician (DE) 4:15
 Auto Rewrite Mode 6:16

- B -

bad block handling = defect management
 <bit> 1:12*
 bit transmission mapping 6:43*
 Bits Stored (DE) 4:15
 block:
 logical 6:11*
 physical 6:11*
 <byte> 1:12*
 byte transmission 6:43*

- C -

CANCEL (ACR) 3:56, 4:22
 <CAPABILITY> 3:27, 52-53
 capability name 4:22*
 CCI = Composite Capability Identification

CFS = Configuration Service
 CFS command 3:64*
 CHANGESTATE (HLF) 3:13, 14*, 19*, 31
 CHANGESTATE/SPI 3:31, 36, 45*, 57, 59, 61, 4:11-12*, 28-30
 channel number selection 5:4*
 <CLASS> 3:21-22, 26, 29-35, 37-42, 51, 54, 61, 63, 4:22
 CLASS (DE) 4:19, 30
 Close Channel Request 5:4
 Coarse Capabilities (DE) 3:15, 27, 52, 4:21*, 23
 command:
 Data Element types 4:28-29*
 mapping (SPI) 3:56*
 name format (SPI) 4:22*
 refusal 3:37*
 request 3:28*
 response 3:28*
 shadow group 3:36, 4:17*, 19*, 20
 Communication Interconnect 1:2*, 5:2*
 <COMPLETION-STATE> 3:55
 COMPLETION-STATE (DE) 4:19, 30
 Composite Capability Identification 4:22-23*
 compression 4:6*, 9, 27*
 Compression Algorithm (DE) 4:26*
 Configuration Service 1:2*, 2:15-16, 3:10, 52-53, 57, 64*
 CONFIGURE (HLF) 3:13, 15*, 27*, 34*, 63
 CONFIGURE/SPI 3:34, 36-37, 52*, 53, 57, 64, 4:11-12*, 21*, 23, 28-30
 CONFIGURE/SPI-RESPONSE:
 "COARSE" 3:52*, 64
 "FINE" 3:52*, 64
 COPY (HLF) 3:13, 14*, 15, 21*, 26, 30-31, 32-34*
 COPY/SPI 3:32-34, 36, 51*, 57, 59, 4:11-12*, 28-30
 <COUNT> 3:23, 25, 49
 COUNT (DE) 4:19, 31
 Country Code 4:3*, 13*

- D -

Data Element 1:2*, 4:2
 description 4:17*
 for command shadow group 4:19
 for DESC 4:15-16
 number actual 4:6-7
 number effective 4:6-7
 number ranges 4:17*
 SPI, special use of 4:18
 Types 4:17*
 Data Object 1:3*, 2:16*, 3:6*,
 4:2, 5, 6:6, 45
 access method 6:8*
 classification 2:2*, 16, 3:6*
 conformance rules 3:6, 7*
 Description 1:3*, 3:7-8, 4:13,
 14*
 Entry 6:35*
 format (optical disk) 6:10, 31
 identification 2:17, 3:8,
 4:13*, 6:6
 Identifier (OMO_ID) 6:6-7, 35
 IMS-known 2:16*, 16-17, 3:5, 8,
 13, 18*, 19-20, 21*, 23, 58,
 60
 IMS-unknown 2:16-17, 3:5, 8,
 13, 18*, 19-20, 21*, 23, 58,
 60, 62
 location 2:16, 3:9*
 modification 3:8*
 modification history 3:8*, 4:25*
 Pointer 6:35*
 State 2:16, 17*
 State labels 2:17
 structure 2:11
 Subtype (DE) 4:8*, 15-16, 25
 Type 2:17*, 3:5
 Type classification 4:10*
 Type (DE) 4:8*, 10*, 15-16, 19,
 25, 31, 6:21-22, 36
 <DATA-OBJECT-SIZE> 3:38
 DATA-OBJECT-SIZE (DE) 3:35, 4:19,
 20*
 Data Set:
 Subtype (DE) 4:15-16, 18
 Type 2:17, 3:7*
 Type (DE) 4:15-16
 <Date> 1:13*
 DATE 6:25*, 26
 DCC = Data Country Code
 <decimal-char> 1:12*
 Defect Management 6:16*, 17
 defect management 6:6, 11*, 24

DESC:

= Data Object Description
 class 4:14*
 Descriptor (DE) 4:25
 elements 3:23, 44, 4:14*
 fixed 4:14*
 for Data Object Type Folder
 4:16*
 for Data Set Type Graphics 4:16*
 for Data Set Type Image 4:15*
 for Data Set Type Text 4:16*
 free-formatted 3:8, 4:14*
 information 3:25
 <DESC> 3:18, 44
 <DESC-SPECIFICATION> 3:38, 40, 44
 <DESTINATION> 3:16, 21, 26, 30,
 35, 51
 DESUBMIT/SPI 3:31, 36, 46*, 47,
 57, 59-60, 4:11-12*, 28-30
 DIALOG (ACR) 3:56, 4:22
 Direct Read During Write 6:11
 Directory Reference Table 1:3*,
 6:10*, 12, 18, 24-26*, 28,
 31-32, 37, 45
 Pointer 6:18*, 19, 24*, 25
 Pointer List 6:25-26*, 45
 Size 6:24
 directory (optical disk):
 reconstruction 6:3
 structure 6:27*
 update procedures 6:9
 DIR_REF_TAB = Directory Reference
 Table
 disk layout:
 and updating 6:39-41*
 with Optional Directory
 Structure 6:41*
 without Optional Directory
 Structure 6:40*
 Distance to Next Secondary Key
 Value 6:37, 38*, 45
 DO = Data Object
 DO# format 4:14*
 domain:
 outside-SPI 3:56*
 SPI 3:56*
 DRDW = Direct Read During Write
 Drive Manufacturer ID 6:16*, 17
 DRT_PTR_LST = Directory Reference
 Table Pointer List
 DST_NVAL = Distance to Next
 Secondary Key Value

- E -

ECHO (ACR) 3:56, 4:22
 Edition 1:9*
 Element 3:24*, 4:24*
 ELEMENT NO 6:38
 element number actual 4:6*, 20*
 element number effective 4:6*, 20*
 <Element-nr> 1:13*, 6:8, 38
 <Enum-16> 1:14*
 Enumerated Value 4:4*
 escape mechanism 4:4*
 SPI 4:30-31*
 escape value 4:4*
 <Even Integer> 3:24*, 4:24
 Export Service 1:3, 2:15-16,
 3:10, 57, 63*
 EXS = Export Service
 EXS commands 3:63*

- F -

FIND (ACR) 3:15, 56, 4:22
 "FINE" 4:22*
 Fine Capability (DE) 3:15, 4:21*,
 23
 floating point number 1:13*
 Folder 4:9
 Group Elements 4:27*
 Type (DE) 4:27, 31
 FORGET (HLF) 3:13, 14*, 18, 20*,
 22, 31
 <FORGET-LAST-COPY> 3:20*, 46, 60
 FORGET-LAST-COPY (DE) 4:19, 30
 FORGET/SPI 3:31, 36, 46*, 48*,
 57, 59-61, 4:11-12*, 20*,
 28-30
 Forgotten Status (DE) 4:19-20
 Format (DE) 4:21*, 23*
 Formatted Disk Capacity 6:16*, 17
 Free Information 6:16*, 17, 18*,
 19, 23*, 26
 Length 6:16*, 17, 18*, 19, 23*

- G -

GCBEGIN/SPI 3:35-36, 54*, 57, 60,
 62-63, 4:11-12*, 22, 28-30
 GCEND/SPI 3:35-36, 55*, 57, 60,
 62-63, 4:11-12*, 22, 28-30
 GET (ACR) 3:14, 30, 4:22
 global index 1:11
 Group 1:3*, 3:24*, 4:24*
 length 4:17

number ranges 4:17*
 GROUP NO 6:38
 GROUPCOPY (HLF) 3:13, 15*, 26*,
 35*, 63
 session 3:15, 38, 40-41, 54-55,
 4:28
 <Group-nr> 1:13*, 6:8, 38

- H -

hard copy:
 output 3:10
 print 3:26, 36*
 unit 1:7
 HEADER 6:32*, 34, 35*, 37, 38*
 hexadecimal number convention
 1:12*, 4:3*
 High Level Function 1:3*, 2:11,
 3:11*, 13*, 16*
 HLF:
 = High Level Function
 argument conventions 3:16*
 description format 3:16*
 "initiator" 3:29*
 "originator" 3:29*
 Translation Table 3:28-35*

- I -

<Ident> 1:14*
 Identifier, Off-Line (OMO_ID) =
 Off-Line Media Object
 Identifier
 Identifier, SPI 1:3*
 Identifier, Unique 1:3*, 3:7, 8*,
 56*, 4:13*
 identifying information elements
 4:25*
 IE = Imaging Equipment
 IE# format 4:13*
 image 1:3*
 exchange format 1:3*, 3:3
 series 1:4*
 storage 1:7
 Image:
 coding 4:6*
 (DE) 4:15-16, 18, 6:7, 21, 36
 Presentation Elements 4:26*
 Image Management Service 1:4*,
 2:15*, 17, 3:5, 8, 10, 13-15,
 44-46, 49, 51, 57, 58*, 4:14
 Data Object registration 3:5,
 13-14*, 17-18*

Imaging Equipment 1:4*, 7, 3:3,
5, 16, 4:13
logical model 3:11*
main functions 2:7
non-SPI-conforming 2:3, 3:12*
SPI-conforming 2:3, 3:6, 12*
IMS = Image Management Service
IMS commands 3:59*
<IMS-FLAG> 3:21-22, 26, 29-30,
35, 38-42, 51, 54, 60-61
IMS-FLAG (DE) 4:19, 30
Index Area 1:4*, 6:10*
<INITIATOR> 3:38-42, 44-46,
48-49, 51-52, 54-55
Institution Name 6:16*, 17
Institutional Department (DE)
4:15, 21*
<Integer> 3:24*
<integer-X> 1:13*
"ISSUE" 4:22*

- K -

Key, Primary 1:4*, 5, 6:3, 6-9*,
19*, 27, 45
Key, Secondary 1:4*, 6, 6:3, 8*,
9-10, 30*, 45
<Keyword> 1:13*
keyword 1:4*

- L -

Label:
Creation Date 6:16*, 17
Creation Time 6:16*, 17
ID 6:16*, 17, 18*, 19, 23*
No 6:16*, 17, 18*, 19, 23*
Version 6:18*, 19, 23*
Version No 6:16*, 17
LD 1200 Laser Drive 6:13*, 16
Length of a Secondary Key Value
6:37, 38*, 45
Length of Object Identifier and
Pointer List 6:38, 39*, 45
Length of the Directory Pointer
List 6:25-26*, 45
LGT_DPL = Length of the Directory
Pointer List
LGT_OIPL = Length of Object
Identifier and Pointer List
<LOCATION> 3:16, 20, 23, 30,
46-47, 49
Location List (DE) 4:19, 20*

logical address 2:7, 4:18*
of an IE 3:9, 16

- M -

Mandatory Directory 1:4*, 6:8*
mapping 6:32*
Structure 6:9, 27*, 45
manufacturer-specific Data
Elements 4:5*
<MAX-RESPONSES> 3:23, 25*, 34, 49
MAX-RESPONSES (DE) 4:19
MDIR = Mandatory Directory
Structure
media:
format 2:18
layout and block numbering
6:10-11*
physical characteristics 6:5*,
13*
recording characteristics 6:5*,
13*

message 1:4*, 2:11, 3:5, 28*, 4:5
modality 1:4*, 7
Modality (DE) 4:15
Modality Type (DE) 4:13, 21*, 31
Modality-Specific:
Viewing Station 1:4*, 7
Workstation 1:5*, 7
Modified Image:
Date (DE) 4:15, 6:22, 36
Identification (DE) 4:15, 6:21,
36
Time (DE) 4:15, 6:22, 36
MOVE (ACR) 3:14, 30
Move Destination Service (DE) 4:31
<MOVE-DESTINATION> 3:42, 62
MOVE/SPI 3:31-34, 36, 42*, 43,
51, 57, 59-60, 62, 4:11-12*,
22, 28-30

Multi-Modality:
Viewing Station 1:5*, 7
Workstation 1:5*, 7
<MY-CAPABILITIES> 3:27*, 52

- N -

<Name> 1:13*
name, logical = logical address
<name-sep-char> 1:12*
NEMA 1:5*
Network Interface Equipment 1:5*,
2:7-10
Network Interface Unit 1:5*

Next Free Block Pointer 6:25-26*,
45
NFB_PTR = Next Free Block Pointer
NIE = Network Interface Equipment
NIU = Network Interface Unit
NOB = Number of blocks
NOB_OBJ = Number of Blocks of the
Object
NOB_OBJ_DIR = Number of Object
Directory blocks
NOB_PKEY_DIR = Number of Primary
Key Directory blocks
NOB_SKEY_DIR = Number of
Secondary Key Directory blocks
NO_OBJ_ENT = Number of Object
Directory Entries
NO_PKV_ENT = Number of Primary
Key Value Entries
NO_SKD = Number of Secondary Key
Directories
Number of blocks 6:32*, 35*, 38*,
45
Number of Blocks:
of Directory Reference Table
6:18*, 19, 24
of the Object 6:35*, 45
Number of Object Directory:
blocks 6:32-33*, 45
Entries 6:34, 35*, 45
Number of Primary Key:
Directory blocks 6:25-26*, 45
Value Entries 6:32*, 45
Number of Secondary Key:
Directories 6:25-26*, 45
Directory blocks 6:25-26*, 45
<NUM-OBJECTS> 3:54
NUM-OBJECTS (DE) 4:19, 21*

- O -

OBJ_DIR = Object Directory
OBJ_DIR_PTR = Object Directory
Pointer
Object Directory 1:5*, 6:27-28,
29*, 34*, 45
contents 6:34*
Entry meanings and formats 6:35*
Pointer 6:28-29, 32-33*, 34, 45
Pointer logical structure 6:29*
Object Entry 6:34, 35*, 45
List 6:30-31, 37-38, 39*, 45

Object Identification 1:5*
Object Identifier 6:21*, 35*, 38,
39*, 45
and Pointer List 6:38, 39*, 45
Descriptor 6:18*, 19, 21-22*,
35-36
Element 6:35-36*
Element "#n" 6:36*
Element "#n" Length 6:21-22*
Element "#n" Name 6:21-22*
ordering 6:36
Object Pointer 6:29, 31, 35*,
37-38, 39*, 45
OBJ_ENT = Object Entry
OBJ_ENT_LST = Object Entry List
OBJ_ID = Object Identifier
OBJ_IDP_LST = Object Identifier
and Pointer List
<OBJ-n-UID> 3:26, 35
OBJ_PTR = Object Pointer
<Odd Integer> 3:24*, 4:24
ODIR = Optional Directory
Structure
off-line media 2:18, 3:10, 63,
6:2*
compatibility requirements 6:3*
data representations 6:43*
Off-Line Media Object Identifier
6:6-7*, 27, 31*, 37, 45
Flag 6:25-26*, 31, 37*, 45
structure 6:6
off-line status 3:27*
off-line storage 1:7
OMO_FLG = Off-Line Media Object
Identifier Flag
OMO_ID = Off-Line Media Object
Identifier
on-line status 3:27*
on-line storage 1:7
Open Channel Request 5:4
<opt-element-data> 1:14*
<opt-element-nr> 1:14*
optical disk 2:18, 6:3, 10, 13*
<opt-info> 1:14*
Optional Directory 1:5*, 6:8*,
10, 37*
alternative forms 6:31*
contents 6:37*
layout 6:30*
mapping 6:37*
Structure 6:9, 30*, 45
<OPTIONAL-ELEMENTS> 3:18, 44
<opt-length> 1:14*

Organ (DE) 4:15-16, 26*
 OSI Model 2:11, 5:2
 vs SPI 2:12-13*, 14
 outside-SPI domain 3:56*

- P -

PACS 1:5*
 PACS console 1:5*, 7
 PACS# format 4:13*, 18*
 PACS-IE Ident 4:18*, 21*
 Patient:
 Age (DE) 4:15-16
 Birthdate (DE) 4:15-16, 6:19, 33*
 Identification (DE) 4:15-16, 6:20, 33*
 Information Elements 4:26*
 Maiden Name (DE) 4:15-16
 Name (DE) 4:15-16, 6:19, 33*
 Sex (DE) 4:15-16, 6:20, 33*
 PBS = Public Storage Service
 PBS commands 3:60-61*
 permanent storage 3:9, 14-15, 20, 45, 6:2
 physical link 2:11, 18*
 Pixel Shadow Group Elements 4:27*
 PKEY = Key, Primary
 PKEY DIR = Primary Key Directory
 PKEY DIR PTR = Primary Key Directory Pointer
 PKEY WORD = Primary Key Word
 PK_VAL = Primary Key Value
 PK_VAL_ENT = Primary Key Value Entry
 Pregnancy (DE) 4:15, 26*, 31
 primary access method 6:8*
 Primary Key Descriptor 6:8*, 18-19*, 19, 33
 Primary Key Directory 6:19-20*, 24, 27, 28*, 32*, 45
 data sequence 6:32*
 Entry meanings and formats 6:32-33*
 logical structure 6:28*
 Pointer 6:25-26*, 28, 32, 45
 Primary Key Value 6:19*, 28, 32-33*, 38, 39*, 45
 Element 6:33*
 Entry 6:32*, 45
 ordering 6:34
 Primary Key Word 6:33*, 45
 Element 6:33*
 <print-char> 1:12*

private extensions 4:8
 Private Service 1:5*, 2:15-16, 3:10, 57, 62*
 Private Storage Service 3:49
 Procedure Description (DE) 4:15-16
 PRS = Private Service
 PRS commands 3:62*
 Public Storage Service 1:5*, 2:15-16, 3:10, 48-49, 51, 57, 58*, 60*

- Q -

Query:
 Elements (DE) 4:21*, 24*
 Match Return (DE) 4:21*, 24*, 31
 Operators (DE) 4:21*, 24*, 31
 Template (DE) 4:19, 21*
 QUERY (HLF) 3:13-15, 23-25*, 34*
 response "COARSE" 3:15, 27*
 response "FINE" 3:15, 27*
 syntax 3:24*
 <TEMPLATE> argument 3:23, 24*
 QUERY/SPI 3:34, 36, 49*, 50, 57, 59-62, 4:11-12*, 20*, 28-30
 template example 4:21*

- R -

Radiologist (DE) 4:15-16
 READY/SPI 3:31-34, 36, 38*, 39-42, 51, 57, 60, 62-63, 4:11-12*, 22, 28-30
 reassign mode (optical disk) 6:11*
 <RECEIVER> 3:38-52, 54-55
 Referring Physician (DE) 4:15
 Report Date (DE) 4:16, 25*
 Report Time (DE) 4:16, 25*
 Reserved entry 6:32, 35, 38
 Reserved space 6:16*, 17-19, 23
 Reset Channel Request 5:4
 <RESPONSES-FLAG> 3:27, 52
 RESPONSES-FLAG (DE) 4:21*, 23, 31
 Resulting Diagnosis (DE) 4:15-16, 25
 <RETURN-INFO-FLAG> 3:23*, 25*, 49
 RETURN-INFO-FLAG (DE) 4:19, 31
 rewrite mode (optical disk) 6:11*

- S -

SCI = Standard Communications Interconnect
 SCSI 6:13, 43

secondary access method 6:8*
 Secondary Key Directory 6:37*, 38, 45
 contents 6:37*
 Entry meanings and formats 6:38-39*
 Pointer 6:25-26*, 37, 45
 Secondary Key Value 6:30*, 31
 Secondary Key Word 6:8*, 25*, 26, 30*, 37, 38*, 45
 SEND (ACR) 3:14, 4:22
 SEND/SPI 3:31-34, 36, 40-41*, 42, 51, 57, 60-63, 4:11-12*, 22, 28-30
 Series (DE) 4:15-16, 18, 6:7, 21, 36
 <SERVICE> 3:35-36, 38-42, 49, 51, 54-55
 Service 1:6*
 (DE) 4:19, 31
 <SESSION-ID> 3:31-34, 35*, 38-42, 51, 54-55, 63
 SESSION-ID (DE) 4:19
 <Shadow Owner Code> 3:24*, 4:24*
 <sign-char> 1:12*
 SKEY = Key, Secondary
 SKEY DIR = Secondary Key Directory
 SKEY DIR PTR = Secondary Key Directory Pointer
 SKEY WORD = Secondary Key Word
 SK_VAL = Value for a Secondary Key Word
 SKV_LGT = Length of a Secondary Key Value
 <SOURCE> 3:16, 21, 30, 51
 <SOURCE-n> 3:26, 35
 Space Limit 6:14, 16*, 17
 SPI:
 = Standard Product Interconnect adapter 2:9-10, 21
 document conventions 1:8*
 document numbering 1:8*
 document set 1:8*
 Document 6 Version No 6:16*, 17
 domain 1:6*, 3:56*
 Enumerated Values 4:30-31*
 environment 2:7*
 Error Status 3:39, 41, 47, 53-54, 4:32*
 extensions to ACR-NEMA 3:11*, 4:2
 goals 2:2*, 5

Recognition Code (DE) 3:36*, 4:4-5*, 8, 19, 24*, 30*
 Release 1:9*
 Release 1: 4:11-12, 30
 requirements 2:3*
 scope of compatibility 2:5-10*
 Status Code (DE) 4:30
 Status Codes 4:32-34*, 35-37
 terminology 1:2
 Version 1:8*
 Volume Label 1:6*, 6:10*, 45
 Volume Label entry meanings 6:17*
 SPI VOL LAB = SPI Volume Label
 Standard Communications Interconnect 5:2*
 Standard Product Interconnect 1:6*, 2:2*
 <STATE> 3:19, 45*
 STATE (DE) 4:19, 20*
 State 3:9, 14
 "ARCHIVED" 2:17, 3:9*, 4:20
 information 2:17*, 3:9*
 private 2:17*, 3:9*
 "REPORTED" 2:17, 3:9*, 4:20
 SPI-defined 2:17, 3:9*
 <State_label> 3:19, 24*, 4:20*
 Station Identification (DE) 4:15-16, 21*, 6:21, 36
 Status Code (DE) 4:19
 Study:
 Date (DE) 4:15-16, 6:21, 36
 (DE) 4:15-16, 6:7, 21, 36
 Time (DE) 4:15-16, 6:21, 36
 SUBMIT (HLF) 3:13, 18*, 31
 SUBMIT/SPI 3:31-32, 36, 44*, 57, 59-61, 4:11-12*, 28-30
 syntax 1:10*

- T -

<TARGET> 3:23, 25, 27, 34
 <TARGET-SUBSET> 3:23
 <TEMPLATE> 3:16, 23, 24*, 49-50
 terminal symbols 1:10*
 <Time> 1:13*
 TIME 6:25*, 26
 <TOTAL-SIZE> 3:35, 54, 4:20*

<TYPE> 3:38

Type, Data Object:

Compressed Image 4:9-12*
 Folder 4:9-12*
 Graphics 2:17, 3:7*, 4:10-12*
 Identifier 4:10-12*
 Image 2:17, 3:7*, 4:8, 10-12*
 Null 4:8, 10-12*
 Other 2:17, 3:7*, 4:8*
 Private Graphics 2:17, 3:7*
 Private Image 2:17, 3:7*, 4:9
 Private Text 2:17, 3:7*
 Text 2:17, 3:7*, 4:8, 10-12*
 TYPE, directory 6:32*, 35*, 38*

- W -

Wait Time (DE) 4:19, 20*
 workstation 1:6*, 7
 WORM = Write Once Read Many
 Write Once Read Many 1:6*, 6:3,
 11*

- U -

UID = Identifier, Unique

<UID> 3:18-21, 38, 40, 42, 44-48,
 51

UID element 3:8*

Unique Identifier 1:3, 6, 3:7
 (DE) 4:13*, 25

User Volume Label 1: 6:8, 10*,
 18*, 24, 33, 35-36, 45

entry meanings 6:19*

User Volume Label 2: 6:10*, 23*,
 45

entry meanings 6:23*

user-specific Data Element 4:5*

USER VOL LAB1 = User Volume

Label 1

USER VOL LAB2 = User Volume

Label 2

- V -

Value for a Secondary Key Word

6:37, 38*, 39, 45

viewing station 1:6*, 7

volume 1:6*

closing 6:12*

layout 6:15*, 40*, 41-42

organization 6:6*, 14-15*

overrun 6:12, 24

physical mapping 6:10*

restoration after crash 6:24

updating 6:12*, 28, 29*, 32*,
 42*

Volume:

Closing Date 6:23*

Completion Status 6:23-24*

ID 6:16*, 17

Label, SPI 1:6*, 6:16*